Chapter 1

DESIGN BACKGROUND AND INTRODUCTION

Introduction

Technology is developed by people to help improve the quality of human lives, and all are using technological advancements in many different ways and one of the ways is security system. When it comes to security systems, biometrics is one of the top choices which have brought significant changes with regard to how people gain access to rooms or establishments. The use of biometrics has changed the security system from what people conventionally use such as passwords, door keys or body parts such as retinal patterns, fingerprints, or voice recognition. Fingerprint recognition is one of the most popular and successful methods used for person identification, which takes advantage of the fact that the fingerprint has some unique characteristics called minutiae. These are points where a curve track finishes, intersect with other track or branches off (Aguilar et. al., 2007).

In this design project, the safety mechanism of the gun will be replaced with fingerprint authentication which can only be accessed by the administrator and registered user.

The fingerprint module that is used in the design is responsible for giving an access to a specific user provided that all allowed fingerprints are stored in the system. The system will then keep a track of all the user's gunshots and record them specifically with the date and time and is sent via short messaging service (SMS).

Customer

APL Security Services Corporation – APL Security Services Corps is a corporate entity registered under the Securities and Exchange Commission (SEC) on March 2007. The agency was put up by individuals who are experienced and knowledgeable in the field of security management. The agency is capable of providing security and safety to its corporate as well as individual clients. The ability to protect life and the ability to prevent loss in any given situation is the corporation's primary concern.

Need

The APL Security Services Corporation needs a device that would prevent misuse of guns and avoid any accidents. As it can seen in the news today, there are several cases of misuse of guns or accidentally pulling the trigger. To prevent events like these to happen again the agency decided to adopt the present design that would improve the safety not only of their clients but also of the weapons that their guards are using.

Solution

With the need to prevent events like these and as requested by the APL Security Services Corporation, the researchers came up with a solution of replacing the safety locks of the firearms with biometrics which uses the fingerprint. The proposed system will replace the existing system with an innovative system that will lessen accidents caused by accidentally pulling the trigger. This will ensure the security of the people.

Objective

The main objective of the project is to create a standalone device that is attached to a gun that asks first for the fingerprint of the registered user before the gun can actually be used. The gun's lock would not be released unless a correct finger print pattern is used. The device will send an alert message once an invalid finger print had been entered. The design is capable of getting the time when the gun is shot, counts its shots and sends it to the database once the firearm is disabled. The time is then sent to a database where they could see whose gun is being used and what time it was b used.

Constraints

In using biometrics fingerprint technology, there are studies showing that among the different biometrics, fingerprint authentication is one of the top if not the top choice to be used for identification systems of devices. Although these studies showed the benefits and competencies in using different kinds of biometrics, the limiting factor for this design project in using biometrics fingerprint technology is the acceptance of the society with fingerprint authentication. Also, the design can only accept two users, one for the administrator and other one for the user. The said design will limited only to a loaded gun. Furthemore, the design is not yet proven to be shock proof, and can sustain the heat of the real gun.

Impact

The design would have a great effect on the safety of the people and stop the abuse and misuse of guns for crime and this will prevent any other accidents that are gun-related. With this innovation, guns are more secured because only the rightful owner has access to this device with its built in fingerprint scanner that would determine if the user is the registered print or not. Also, passing and sharing of guns will not be possible because for every person, there is a unique finger print and only the registered owner can pull the trigger. It can also affect the social relation of the agency to its clients ensuring the safety of their customers and having a more advanced weapon that is more secured than the conventional gun.

Differentiation

This design project is unique since the only people who can access the firearm will be the administrator and the user, compared to the firearms today which is can be used by anyone. In the Biometric Analysis for Firearm Identification with Real-Time Detection for Ballistic Projectile via SMS, the trigger can only be accessed by the registered user of the gun. It sends an alerts message through the SMS once an invalid fingerprint has been identified. It is capable of transmitting the total number of shots, the time and date once the firearm is disabled. The fingerprint scanner allows the user to unlock the gun whenever he wants to use it. Lastly, it can help in the investigation of crimes. While in the Fingerprint Gun Lock/Safe – Gun Vault, the device only serves as

storage of the gun that can only be accessed by a valid fingerprint. Thus, anyone can still use the firearm once the Gun Vault is unlocked.

Benefits

The Biometric Analysis for Firearm Identification with Real-Time Detection for Ballistic Projectile via SMS comes with features that would improve the safety of using firearms. APL Security Services Corporation can improve their weapons through this device. The design would reduce the likelihood of unintentional injuries to children, teenage suicides and homicides and limit the violent acts committed by criminals using stolen guns. It can also protect law enforcement officers from criminals who may grab the firearms during a struggle.

Definition of Terms

Biometrics - consists of methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. It is used as a form of identity access management and access control. It is also used to identify individuals in groups that are under surveillance.

Fingerprint - an impression left by the friction ridges of a human finger. It is the trace of an impression from the friction ridges of any part of a human hand.

PIC Microcontroller –The name PIC initially referred to "Peripheral Interface Controller". PIC are popular with both industrial developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application

notes , availability of low cost or free development tools, and series programming (and reprogramming with flash memory) capability.

Solenoid – In Engineering, the term solenoid may also refer to a variety of transducer devices that converts energy into linear motion. The term is also used to refer to a solenoid valve, which is an integrated device containing an electromechanical solenoid which actuates either a pneumatic or hydraulic valve, or a solenoid switch, which is a specific type of relay that internally uses an electromechanical solenoid to operate an electrical switch.

GSM/GPRS Module –it is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile Communication (GSM) is an architecture used for mobile communication in most of the countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces for computer. GSM/GPRS Modem is a class of modem device that is designed for communication of a computer with the GSM and GPRS network. It requires a Subscriber Identity Module (SIM) card just like mobile phones to activate communication with the network.

Chapter 2

REVIEW OF RELATED LITERATURES AND STUDIES

Biometrics

Biometric authentication refers to the identification of humans by their characteristics or traits. Biometrics is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance. Biometrics identifiers are the distinctive measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological versus behavioral characteristics. Many different aspects of human physiology, chemistry or behavior can be used for biometric authentication. The selection of particular biometric for use in a specific application involves a weighing of several factors. Universality means that every person using a system should possess the trait. Uniqueness means that trail should be sufficiently different for individuals in the relevant population such that they can be distinguished from one another. Permanence relates to the manner in which a trait varies over time. More specifically, a trait with good permanence will be reasonably invariant over time with respect to the specific matching algorithm. Measurability relates to the ease of acquisition or measurement of the trait. In addition, acquired data should be in a form that permits subsequent processing and extraction of the relevant feature sets. Performance relates to the accuracy, speed, and robustness of technology used. Acceptability relates to how well individuals in the relevant population accept the technology such that they are willing to

have their biometric trait captured and assessed. Circumvention relates to the ease with which a trait might be imitated using an artifact or substitute.

Biometric system can operate in two modes. In verification mode, the system performs a one-to-one comparison of a captured biometric with a specific template stored in a biometric database in order to verify if the individual is the person they claim to be. There are three steps involved in person verification. In the first step, reference models for all the users are generated and stored in the model database. In the second step, some samples are matched with reference models to generate the genuine and impostor scores and calculate the threshold. The third step is the testing step. This process may use a smart card, username or ID number to indicate which template should be used for comparison. Positive recognition is a common use of verification mode where the aim is to prevent multiple people from using same identity. In identification mode, the system performs a one-to-many comparison against a biometric database in attempt to establish the identity of an unknown individual. The system will succeed in identifying the individual if the comparison of the biometric sample to a template in the database falls within a previously set threshold. Identification mode can be used either for positive or negative recognition of the person where the system establishes whether the person is who he/she denies to be. The latter function can only be achieved through biometrics since other methods of personal recognition such as passwords, PINs or keys are ineffective. The first time an individual uses a biometric system is called enrollment. During the enrollment, biometric information from an individual is captured and stored. In subsequent uses, biometric

information is detected and compared with the information stored at the time of enrollment. The first block (sensor) is the interface between the real world and the system; it has to acquire all the necessary data. Most of the times, it is an image acquisition system, but it can change according to the characteristic desired. The second block performs all the necessary pre-processing: it has to remove artifacts from the sensor, to enhance the input, to use some kind of normalization, etc. In the third block, necessary features are extracted. If enrollment is being performed, the template is simply stored somewhere. If a matching phase is being performed, the obtained template is passed to a matcher that compares it with other existing templates, estimating the distance between them using any algorithm. The matching program will analyze the template with the input. This will then be output for any specified use or purpose.

In the article entitled "Biometrics", Biometrics was defined as an open-minded set of technologies based on the measurement of some unique physical characteristics of an individual for the purpose of identifying an individual or verifying identity which cannot be borrowed, stolen, or forgotten. This technology measures the individual's unique physical or behavioural characteristics to recognize or authenticate their identity. Behavioural characteristics include signature, signature dynamics, voice, lip movement, keystroke analysis, and gait. Physical characteristics include hand geometry, retina, iris, facial characteristics and fingerprints (Dilum Bandara, 2008).

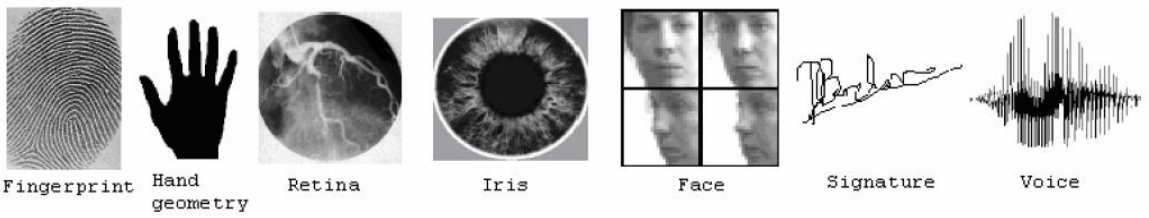


Figure 2.1 Various Biometrics Technologies

Fingerprint Recognition

Fingerprint recognition or authentication refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of biometrics used to identify individuals and verify their identity. The analysis of fingerprints for matching purpose generally requires the comparison of several features of the print pattern. These include patterns, which are aggregate characteristics of ridges, and minutia points, which are unique features found within the patterns. It is also necessary to know the structure and properties of human skin in order to successfully employ some of the imaging technologies. The three patterns of fingerprint ridges are the arch, loop, and whorl. In arch the ridges enter from one side of the finger, rise in the center forming an arc, and then exit the other side of the finger. In loop the ridges enter from one side of a finger, form a curve, and then exit on that same side. Finally, in whorl ridges form circularly around a central point of the finger.

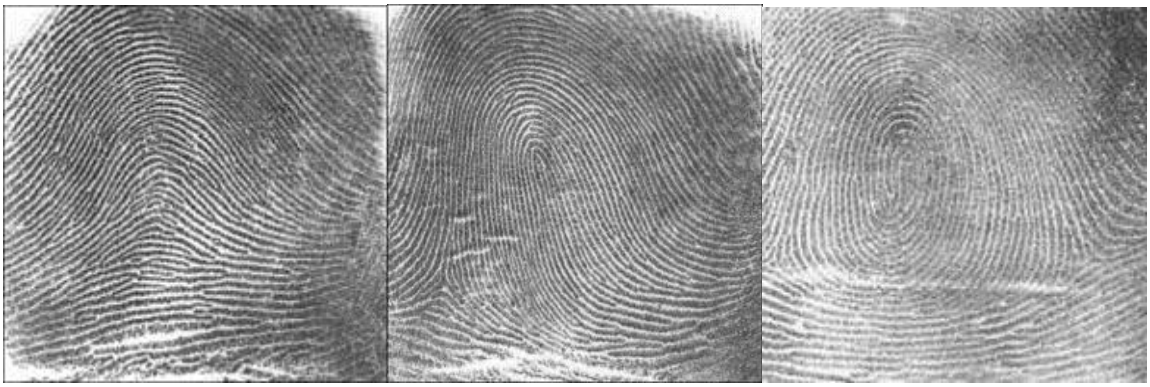


Figure 2.2 Arch, Loop and Whorl Pattern

Minutia Features

The major minutia features of fingerprint ridges are ridge ending, bifurcation, and short ridge (or dot). The ridge ending is the point at which a ridge terminates. Bifurcations are points at which a single ridge splits into two ridges. Short ridges (or dots) are ridges which are significantly shorter than the average ridge length on the fingerprint. Minutia and patterns are very important in the analysis of fingerprints since no two fingers have been shown to be identical.

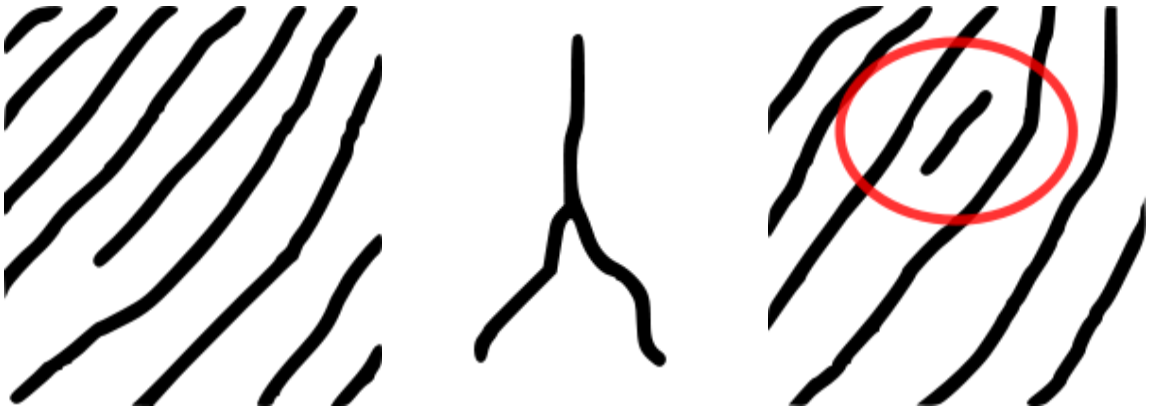


Figure 2.3 Ridge ending, Bifurcation, Short Ridge (Dot) Pattern

Fingerprint Sensors

A fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted features) which is stored and used for matching. Fingerprint is one of the most common biometrics used in the field of security which still faces challenges that present acceptance. These challenges include threat of identity theft which has been validated as a realistic vulnerability. in the study by Palka and Hamilton (2009) in the "Fingerprint Readers: Vulnerabilities to Front- and Back- end Attacks", the researchers stated that the technology used for sensors and fingerprint processing has matured but vulnerabilities still persist as described in their paper.

Access Control System with High Level Security Using Fingerprints

The article presented a biometric-based applications that guarantee resolving numerous security hazards. As a method of preserving privacy and the security of sensitive information, biometrics has been studied and used for the past few decades. Fingerprint is one of the most widely used biometrics. A number of fingerprint verification approaches have been proposed until now. However, fingerprint images acquired using current fingerprint input devices that have small field of view are from just very limited areas of whole fingertips. Therefore, essential information required to distinguish fingerprints could be missed or extracted falsely. The limited and somewhat

distorted information are detection from them, which might reduce the accuracy of fingerprint verification systems. In the system that verifies the identity of two fingerprints using fingerprint features, it is critical to extract the correct feature information. In order to deal with these problems, compensation of imperfect information can be performed using multiple impressions of enrollee's fingerprints. Additional three fingerprint images are used in enrollment phase of fingerprint verification system. Traditionally, verified users have gained access to their property or service via dozens of PIN/password, smart cards and so on. However, these knowledge from plural fingerprint images on enrollment stage. The proposed enrollment uses multiple fingerprint images to extend enrolled fingerprint image and also guarantee the reliability of each fingerprint image. Figure 2.4 presents a fingerprint verification system, which consists of two phases: enrollment and verification. In the offline enrollment phase, at first, the fingerprint image of an enrollee is acquired and pre-processed. Then, the minutiae are extracted from the raw image and stored as enrolled template. And in the online verification phase, it reads the fingerprint from a claimer, and detects the minutiae information through the same procedure as in the enrollment phase. Then, it estimates the similarity between the enrolled minutiae and the input minutiae.

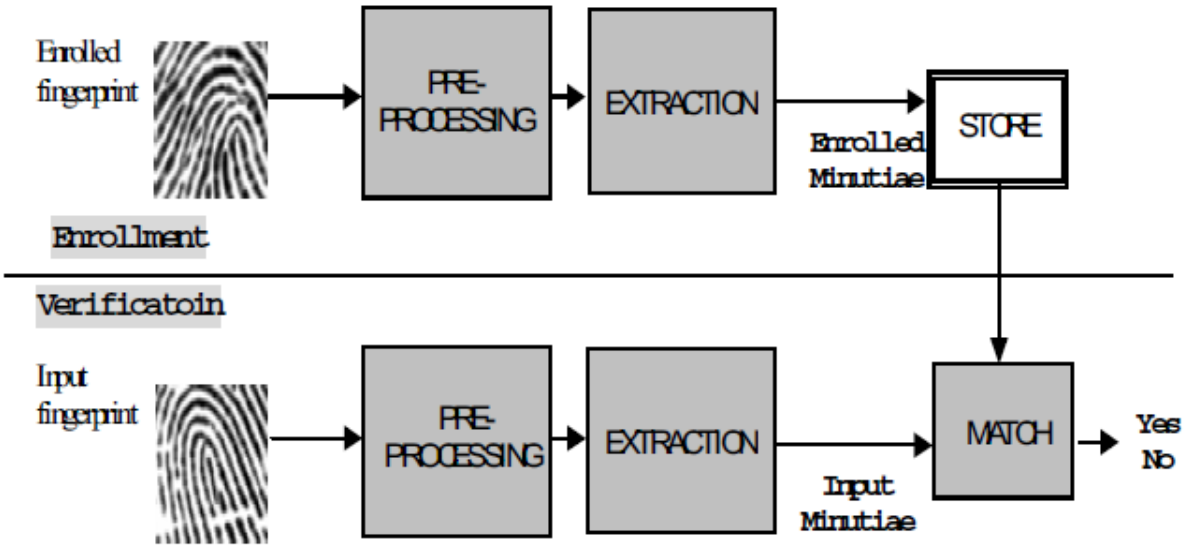


Figure 2.4 User authentication system using fingerprint

As automatic fingerprint identification and authentication systems rely on representing the two most prominent minutiae, a reliable minutiae extraction algorithm is critical to the performance of the system. However, although minutiae are detected through not only extraction process but also false minutiae removal process, it is still possible to detect false minutiae and miss true one. These faults can cause matching failure. In particular, if they occur during enrollment phase and are stored as enrolled template, it will be serious problem, because they will affect the matching phase continuously. The suggested plural fingerprint images on the enrollment phase to discard the false minutiae and compensate the missed minutiae. The system adopts the enrollment using multiple fingerprint images as shown in Figure 2.5 (Gil, Ahn, Pan, Yongwha, 2003).

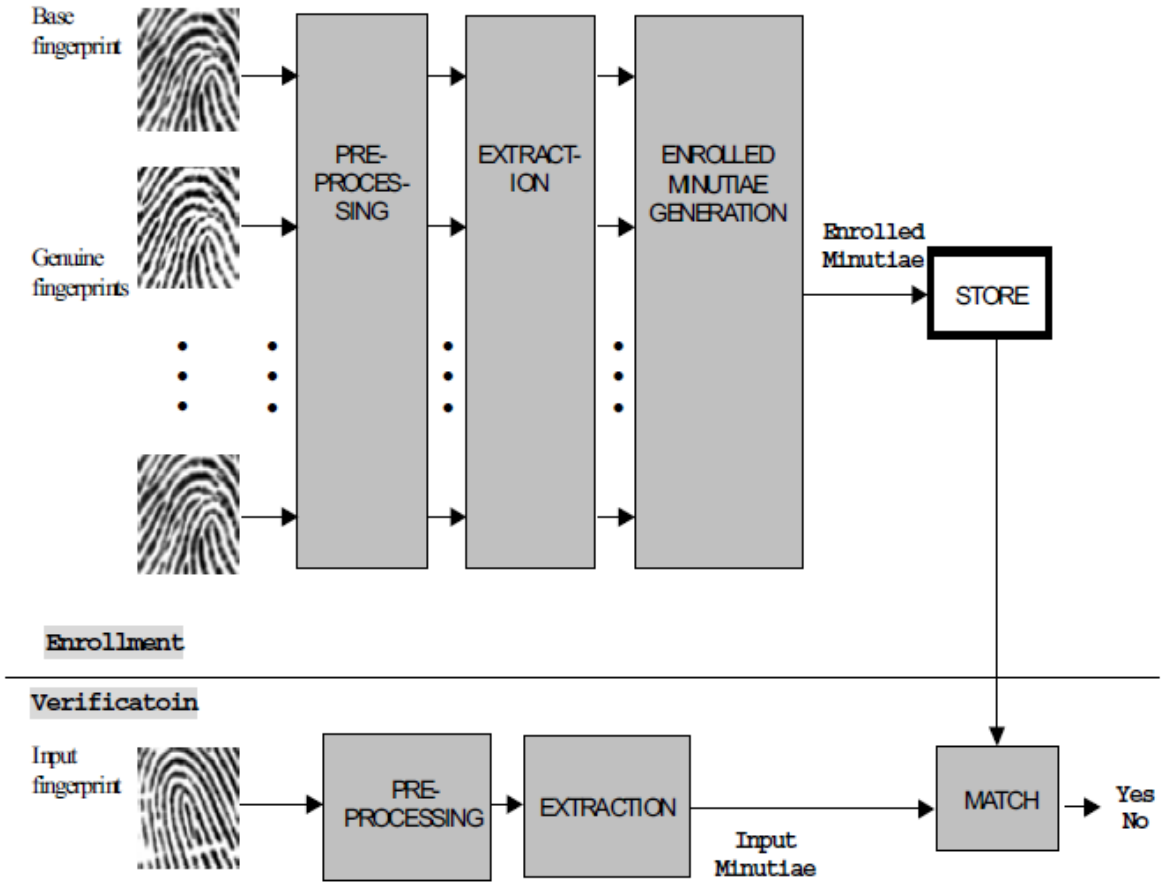


Figure 2.5 User Authentication system that adopts the enrollment using plural fingerprint images

Fingerprint Recognition – Based Access Controlling System for Automobiles

Traditional identifying system of automobiles includes key system, door-controlling system, Ultra High Frequency (UHF) sending/receiving system, alarm system and many more. The paper presented an embedded application system access control system based on fingerprint recognition for automobiles, which is a kind of Automatic

Fingerprint Identification System (AFIS) is introduced. Biometric technologies have great potential in various technical fields such as network security, public security, and financial industry system. Biometrics has grown into an independent industry, whose standardization is making rapid progress with a large number of standards. These standards have focused on various aspects of biometric industrialization, including data formats, application program interfaces, security issues, system profiles, and evaluation methodologies. Access control system based on fingerprint recognition for automobiles is an AFIS. Once the unauthorized key is used to open the door of the automobile, it can alarm through buzzers, and it also makes its lamps flash constantly. The technique presented is very convenient to open the automobile's door with the fingerprint encryption key system. The paper introduces an embedded application system access control system based on fingerprint recognition for automobiles, which is a kind of Automatic Fingerprint Identification System (AFIS).

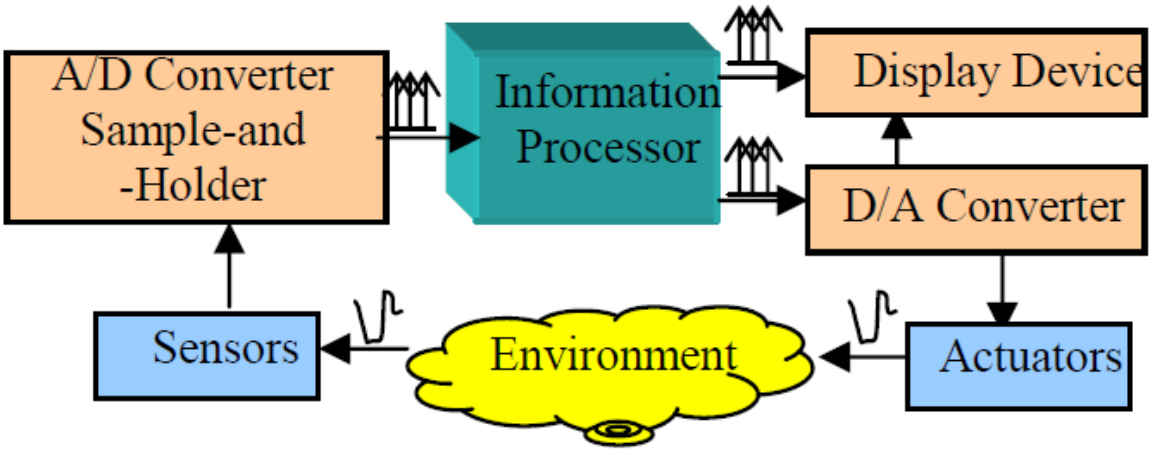


Figure 2.6 Architecture of Embedded System

ATMEL invents fingerprint sensors which can capture slices of the fingerprint when a finger slides over these sensors and put them together to obtain a complete fingerprint image with very low power consumption, wide operating temperature range and full speed. It includes two Key Systems (KSs) and a Door-Controlling System (DCS). One KS built on the door of the automobile is composed of a finger-chip and a circuit connected with DCS and, the other one taken by the driver is constitutive of a finger-chip and an Ultra High Frequency (UHF) sender capsulated in plastic (Zhu & Chen, 2011).

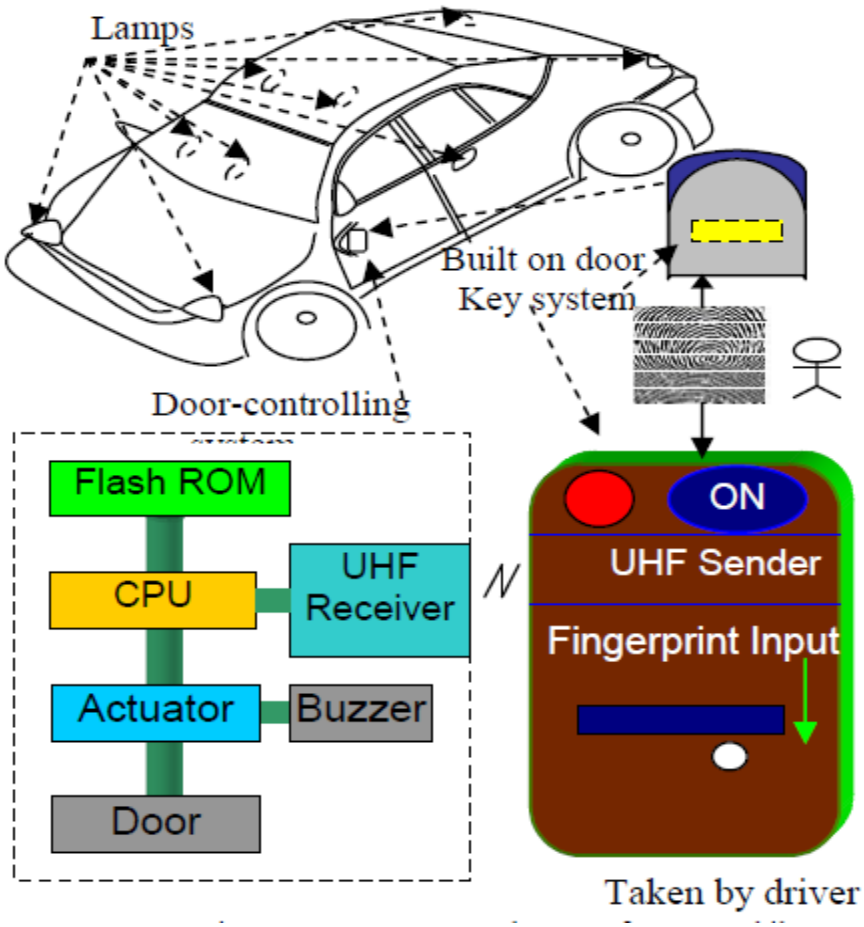


Figure 2.7 Access control system for automobiles

Computerized System for Fingerprint Identification for Biometric Security

AFIS (Automated Fingerprint Identification System) is used for identification of fingerprint images. AFIS is a type of biometric system which uses digital imaging methodology to capture, analyze and store fingerprint data. This fingerprint data can be compared with the database to determine the identity of an individual. The fingerprint images of poor quality need to be enhanced before the feature extraction and feature matching. The article presents an automated approach for fingerprint matching, which involves three steps. First is to enhance the image to remove noise. Second is to extract required feature that is necessary for matching and remove unwanted feature. The final step is matching by using adjacent feature vector technique. The advantage of adjacent feature vector technique is that; it shows a good performance to differentiate matching and non-matching minutiae pairs. AFV (Adjacent Feature Vector) also reduce the computation cost and matching error. The researchers used standard fingerprint image database FVC2004 (Fingerprint Verification Competition). Image Enhancement is used to enhance the image and then features are extracted because they want exact matching so they eliminate unnecessary feature and extract required feature and after this extracted features are matched. This system provides the capable, correct and error free results. Fingerprint is the surface that is present on the top of human fingertip. Every person has unique and unchallengeable fingerprint and due to this fingerprint authentication has been useful in many areas. Fingerprint identification is the process of automatically matching one or more fingerprint images against a

database. Computerized fingerprint identification system is mostly used for security purposes (Batool & Tariq, 2011).

A New High-Level Security Portable System Based on USB Key with Fingerprint

With the help of the USB Key and fingerprint, people have developed different kinds of authentication solutions to protect the confidential information against unauthorized access. The article presents the above solutions to these problems. One is that the sensitive information is not encrypted and the other one is that the environment is insecure. A new high-level security portable system is used to solve these problems. The system is a combination of hardware and software. Based on the USB Key and fingerprint scanner, the plug and play USB 2.0 device was integrated without the driver installation. By rebuilding the syslinux boot loader and customizing the Puppy Linux, a fingerprint based authentication security portable custom-built software environment inside the device was developed. With the help of the system, the users can work at a movable security environment freely, as well as the high-level protection of confidential information. The high level security portable system is designed for extremely high demands on security and confidentiality of electronic information industry market with the biometric security technology system. It is also an on-chip authentication system; all the authentications are run inside the USB Key without the host OS's help. At the same time, they installed the entire software

environment on the USB flash controlled by the USB Key. This software environment can be customized according to the user's demand and the authentication process is at the computer boot stage implemented under the USB mass-storage protocol with the SCSI RBC command. The high level security portable system is designed to be a plug and play USB 2.0 device and can communicate with the host OS which supports the USB device.

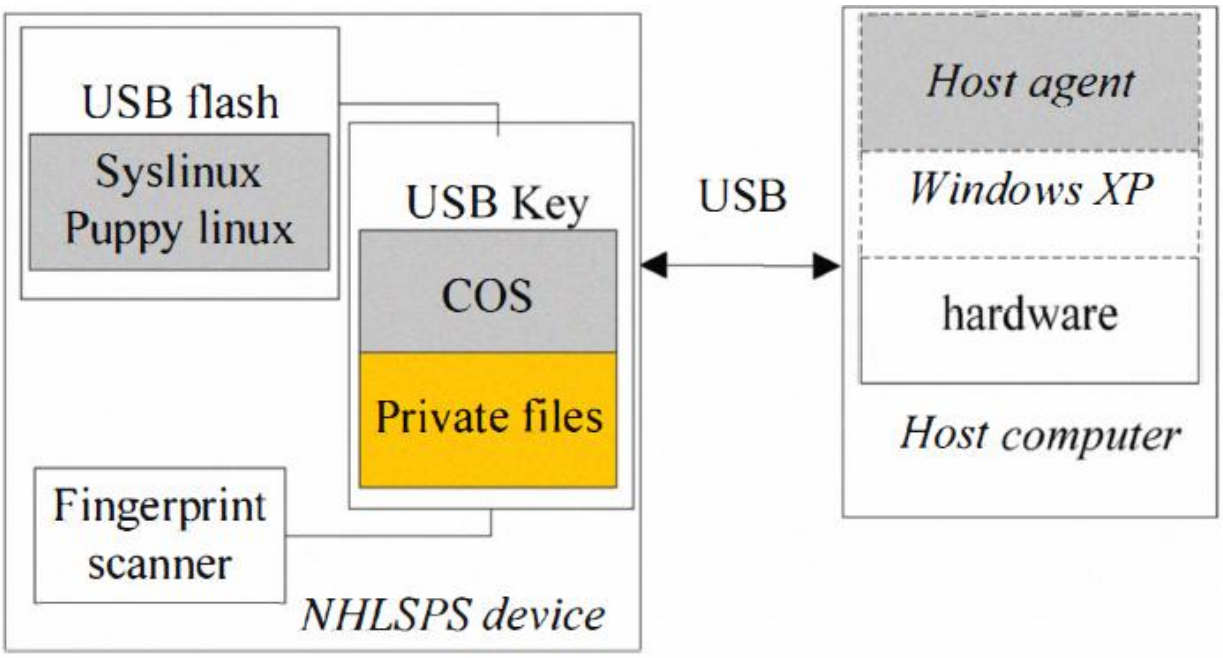


Figure 2.8 High Level Security Portable System Architecture

In the device, the private files such as the private encryption key and fingerprint features value are stored on the on-chip flash. The entire authentication operations carry out instead the device without the help of host computer. The security environment, syslinux and Puppy Linux, is installed on the USB flash partition. After finishing the enrollment, the device can be used with any computer that supports the

USB 2.0 with the BIOS support of booting from removable disks. The article presents a new security integrated system that enables users to work at a movable security environment with good protection of confidential information. The High Level Security Portable System carries out the authentication while system booting, and protects the confidential files by hardware encryption. With the Puppy Linux customization, HLSPS provides the users a high-level security portable work environment (Li & Chen, 2010).

CHAPTER 3

DESIGN PROCEDURES

This chapter gives a detailed discussion on how the step-by-step procedure will be used on the design in order to give the readers the idea on how the prototype has been created. This also helps the readers to easily understand how the group contributed to be able to theorize the development of the design. With the help of the idea and data that were gathered in Chapter 2, the group came up the idea on how the design will be implemented as shown in Figure 3.1.

In this design project, the safety mechanism of the gun will be replaced with fingerprint authentication which can only be accessed by the administrator and registered user.

The fingerprint module that is used in the design is responsible for giving an access to a specific user provided that all allowed fingerprints are stored in the system. The system will then keep a track of all the user's gunshots and record them specifically with the date and time and is sent via short messaging service (SMS).

HARDWARE DEVELOPMENT

BLOCK DIAGRAM

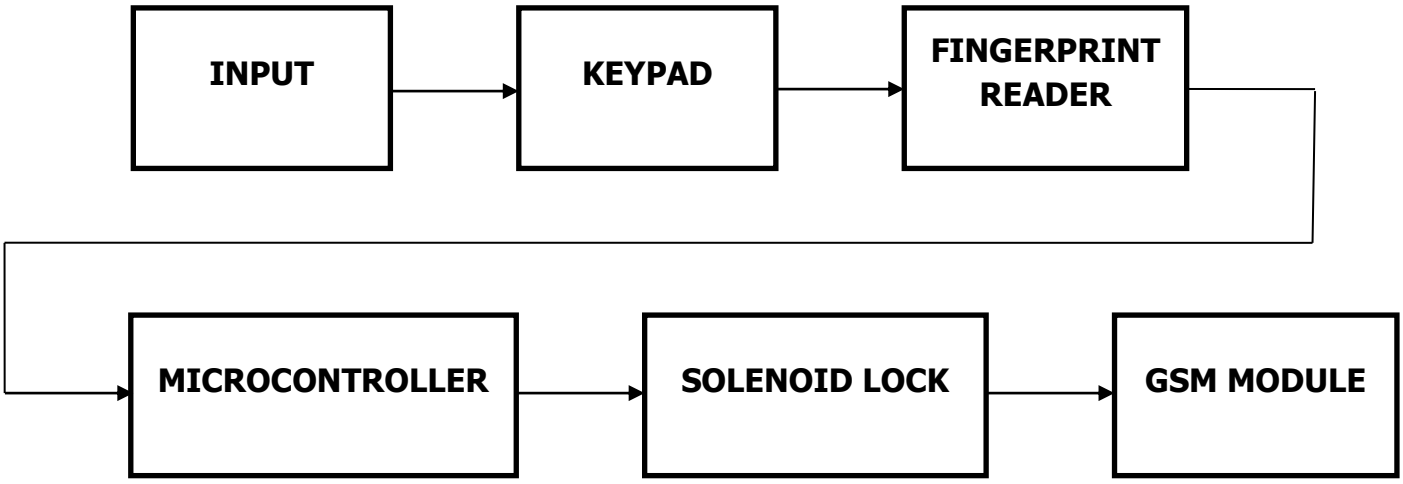


Figure 3.1 Block Diagram of the Design

Figure 3.1 shows the block diagram of the design. To access the fingerprint reader, the pound sign must be pressed on the keypad to turn the fingerprint reader on. It will then validate the fingerprint if it has an access or not. The fingerprint reader only accepts two valid fingerprints, the administrator and user's fingerprint. The microcontroller configures which mode would be given based on the fingerprint that was scanned more recently. In the administrator mode, the user has all the capabilities on doing anything on the system. While in the user mode, once the system detects that the fingerprint entered is for a user, the system would then automatically unlock the solenoid lock. Finally, the system records the activity of the users that are using the

gun. It counts the total number of shots the gun fired, the time and date when the gun was used and it is sent through the GSM module.

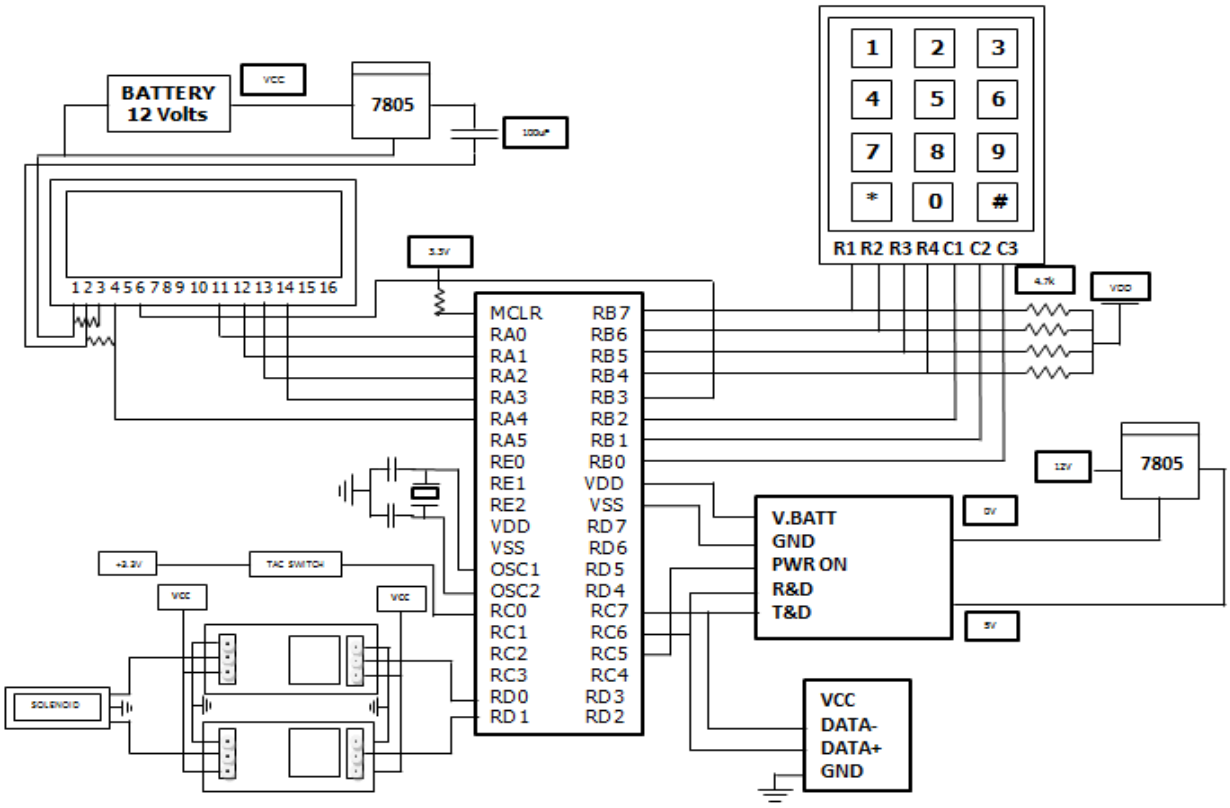


Figure 3.2 Schematic Diagram

Figure 3.2 presents the schematic diagram of the Biometric Analysis for Firearm Identification with Real-Time Detection for Ballistic Projectile via SMS. The schematic diagram is divided into several parts: the Biometric Module, GSM Module, and

the Solenoid lock that is attached to the firearm. The fingerprint reader is the main system of the design that is required to activate the firearm.

The fingerprint reader is the one responsible for collecting users' fingerprints. It is regulated by a 5V output voltage. An oscillator of 20 mega-Hertz is connected to one of the microcontroller's pins for the purpose of timing frequency. A 100 micro-faradcapacitor is connected to the output voltage of the regulator and the ground to filter out the noise coming from the regulator.

The GSM Module is the one responsible in transferring the data gathered by the microcontroller. It is regulated by a 5V output voltage and 100 micro-farad capacitor is connected to the output voltage of the regulator and the ground to filter out the noise coming from the regulator. The relay circuit is connected to the 12V battery that is required to trigger the push/pull solenoid lock.

SOFTWARE DEVELOPMENT

PROGRAM FLOW CHART

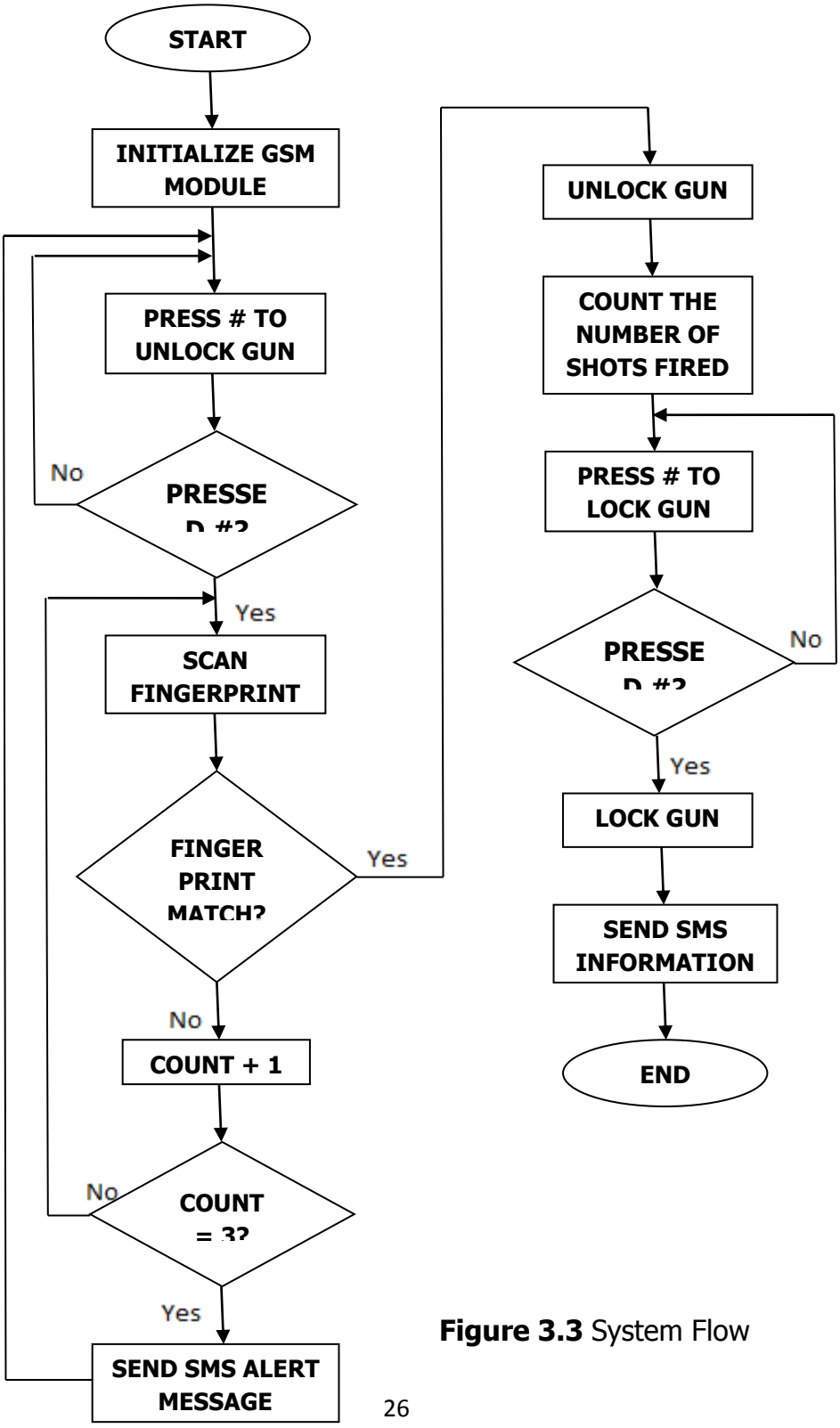
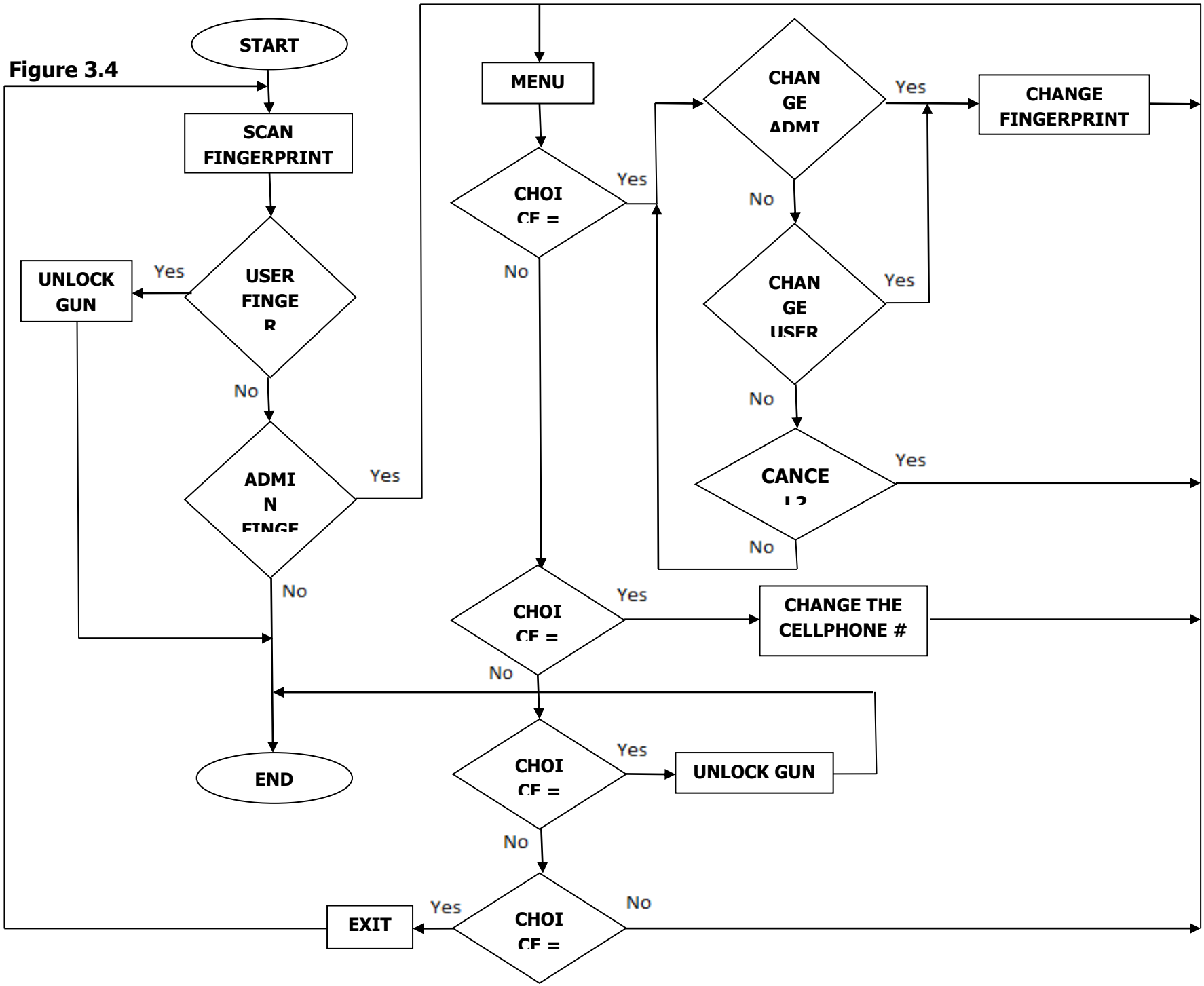


Figure 3.3 System Flow

Figure 3.4


START
SCAN FINGERPRINT
USER FINGE R
Yes
UNLOCK GUN
No
ADMI N FINGE
Yes
No
END
MENU
CHOI CE =
Yes
No
CHAN GE ADMI
Yes
CHANGE FINGERPRINT
No
CHAN GE USER
Yes
No
CANCE L ?
Yes
No
CHOI CE =
Yes
CHANGE THE CELLPHONE #
No
CHOI CE =
Yes
UNLOCK GUN
No
CHOI CE =
Yes
EXIT
No

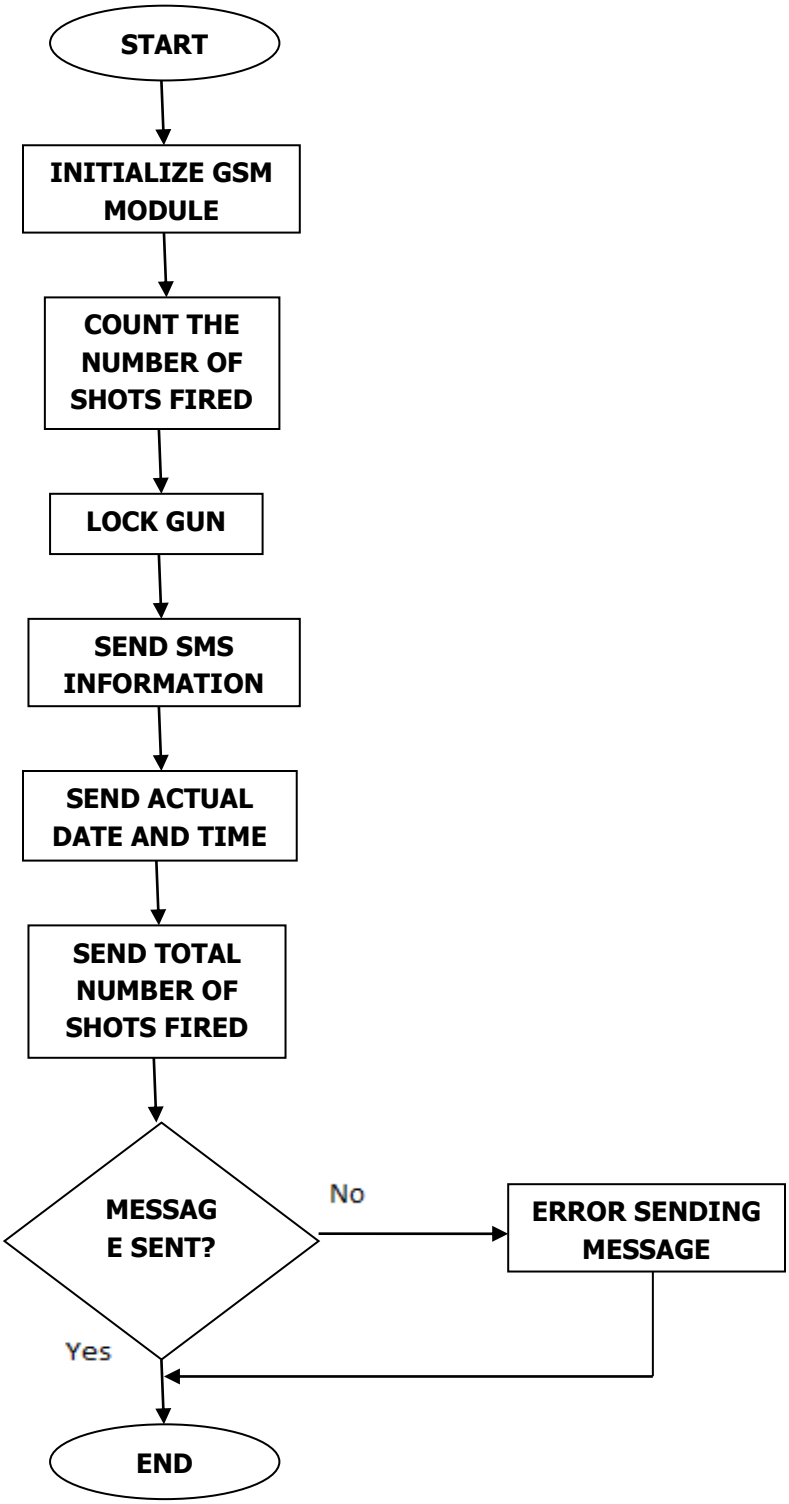


Figure 3.5 GSM Module System Flow

The figures 3.3, 3.4 and 3.5 are the flow charts of the prototype. Figure 3.3 shows the process of the whole system of the Biometric Analysis for Firearm Identification with Real-Time Detection for Ballistic Projectile via SMS. The system starts up when the device is turned on and immediately initializes the GSM Module to acquire signal from the network. The system would wait for the user to press the # sign to send a signal to the device that the user is about to enter a fingerprint for verification. The fingerprint reader would then identify if the scanned fingerprint is enrolled or not. If the fingerprint entered is not enrolled, the system would start an error count that signifies that the fingerprint entered is not enrolled in the system. When the system counts three errors, the system would send a SMS message that will inform that an unauthorized fingerprint if trying to access the gun. If the fingerprint is enrolled, the system will unlock the gun and allows the user to use the gun. The system will count the number of shots fired by the gun. Once the # sign is pressed again, it will lock the gun and send the total number of shots fired, the time and date when the gun was used.

Figure 3.4 presents the two modes of the fingerprint reader, the administrator and the user modes. In the user mode, when the entered fingerprint is verified as user fingerprint the system will automatically unlock the gun. While in the administrator mode, the system will display a menu of the functions that the administrator is capable of carrying out. The administrator is allowed to change the user's fingerprint and also his own fingerprint. It can also change the cellphone number where the message would be sent. Lastly, the administrator mode can also unlock the gun or choose to exit the menu to be able to scan another fingerprint. Figure 3.5 presents the GSM Module system flow, when

the gun is locked, the system will send an SMS message containing the total number of shots fired by the gun together with its real time and date of its use to the cellphone number that is registered in the system. When the system fails to send the message it will display a message informing that the system failed to send the message.

PROTOTYPE DEVELOPMENT

Figure 3.6 GSM Module

The GSM Module is a specialized type of modem which accepts a SIM card and operates over a subscription to a mobile operator. The GSM modem is integrated to the microcontroller that allows the device to communicate to the mobile network and used for sending SMS messages.

Figure 3.7 Biometric Module

The Biometric Module serves as the primary device for the verification of the fingerprints. It is also used to determine the mode of the user of the gun.

Figure 3.8 Microcontroller

The microcontroller unit is in charge of receiving data from the Biometric Module and sends the signal to lock and unlock the solenoid. It is also used to count the number of shots fired by the gun and allows the GSM Module to send the data gathered.

Figure 3.9 Solenoid

The solenoid is used to lock or unlock the gun in the system. It is considered as a output and waits for the signal from the microcontroller.

Figure 3.10 Resistors

The resistors are used in order to balance the flow of current in the circuit. The resistors are used basically for pull-up in order to produce proper input going through the device.

Figure 3.11 Connecting Wires

The connecting wires are used to enhance the range of the components like the solenoid, GSM Module and the Biometric Module. Stranded wires are being used in the circuit since it is more flexible.

COMPONENT	PRICE (PHP)	QUANTITY	PRICE (PHP)
Biometric Module	3800.00	1	3800.00
GSM Module	1800.00	1	1800.00
PIC16F877A	250.00	1	250.00
40 Pins IC Socket	20.00	1	20.00
20mHZ Crystal Oscillator	30.00	1	30.00
2 Pins Connector	10.00	1	10.00
LCD Display	1250.00	1	1250.00
Keypad	120.00	1	120.00
Tac Switch	20.00	1	20.00
4.7 kOhm Resistor	1.25	6	7.50
24pF Capacitor	1.00	2	4.00
100uF Capacitor	2.50	3	7.50
7805 Transistor	12.00	2	24.00
1N4001 Diode	1.00	1	1.00
4 x 6 PCB	100.00	1	100.00
Total			7444.00

Table 3.1 Component Price Listing

Chapter 4

TESTING, PRESENTATION, AND INTERPRETATION OF DATA

Functionality Test

The system can verify whether the scanned fingerprint is registered or not. It also has add and delete fingerprint functions for registered fingerprints and can send text messages with information as regards number of gun shots. The researchers tested the features of the Biometric Analysis for Firearm Identification with Real-Time detection for Ballistic projectile via SMS: Verify Add fingerprints and Real-Time Detection via SMS.

Procedure for Verification Feature:

1. First turn on the switch located at the back part of the firearm.
2. Click the # button.
3. Once the scanner lights up, scan the fingerprint.
4. If the fingerprint scanned is from the admin the status will be admin; if it is from a user its status will be user, if not nothing will happen. After three tries, a text message will be sent to the registered phone number on the system notifying the admin that there is someone unauthorized who used the firearm.

Procedure for the adding feature (Only the administrator has the rights to add a new user to the said firearm):

1. First turn on the switch located at the back part of the firearm.
2. Click the # button.
3. Once the scanner lights up, scan the fingerprint.
4. Once logged in, there will be four choices for the admin: (1) for add a user, (2) change the registered phone number, (3) unlock gun, and (4) exit. Since the device is adding a user, Press (1).
5. Wait for the scanner to light up.
6. Scan the new user's fingerprint three times and wait for the light to turn off.

TRIAL	VERIFICATION	STATUS
1	SUCCESS	ADMIN
2	SUCCESS	ADMIN
3	SUCCESS	ADMIN
4	SUCCESS	ADMIN
5	SUCCESS	USER
6	SUCCESS	USER

7	SUCCESS	ADMIN
8	SUCCESS	ADMIN
9	FAIL	LOGIN
10	FAIL	LOGIN
11	FAIL	LOGIN
12	SUCCESS	ADMIN
13	SUCCESS	ADMIN
14	SUCCESS	ADMIN
15	SUCCESS	USER
16	SUCCESS	USER
17	SUCCESS	ADMIN
18	SUCCESS	USER
19	SUCCESS	ADMIN
20	FAIL	LOGIN
21	FAIL	LOGIN
22	SUCCESS	ADMIN

23	SUCCESS	ADMIN
24	SUCCESS	ADMIN
25	SUCCESS	ADMIN
26	FAIL	LOGIN
27	SUCCESS	ADMIN
28	SUCCESS	USER
29	SUCCESS	USER
30	SUCCESS	ADMIN

Table 4.1 Registered Fingerprint Testing

Based on the testing results on table 4.1, the Biometric analysis for firearm identification with real-time detection for ballistic projectile via SMS design prototype precisely validates if the scanned fingerprint is registered or not in its database. There were some occurrences that the results failed, since the registration of fingerprints depends on how the particular finger was scanned in the Biometric analysis of Firearms as the fingerprint reading gathers three samples of fingerprint template for precision. The researchers consider the misplacement of the specific finger to the fingerprint scanner as the cause of this problem. Furthermore, having three login failures will result in sending of SMS to the cellphone. Further testing about this occurrence will be shown on to the succeeding testings.

Figure 4.1 Fingerprint Scanner Start-up

Figure 4.2 Fingerprint Administrator Mode

Figure 4.3 Fingerprint User Mode

Figure 4.4 Fingerprint Access Denied

TRIAL	VERIFICATION	STATUS
1	ADD USER	REGISTERED
2	ADD USER	REGISTERED
3	ADD USER	REGISTERED
4	ADD USER	REGISTERED
5	ADD USER	REGISTERED
6	ADD USER	REGISTERED
7	ADD USER	REGISTERED
8	ADD USER	REGISTERED
9	ADD USER	REGISTERED
10	ADD USER	REGISTERED
11	ADD USER	REGISTERED
12	ADD USER	REGISTERED
13	ADD USER	REGISTERED
14	ADD USER	REGISTERED
15	ADD USER	REGISTERED

16	ADD USER	REGISTERED
17	ADD USER	REGISTERED
18	ADD USER	REGISTERED
19	ADD USER	REGISTERED
20	ADD USER	REGISTERED
21	ADD USER	REGISTERED
22	ADD USER	REGISTERED
23	ADD USER	REGISTERED
24	ADD USER	REGISTERED
25	ADD USER	REGISTERED
26	ADD USER	REGISTERED
27	ADD USER	REGISTERED
28	ADD USER	REGISTERED
29	ADD USER	REGISTERED
30	ADD USER	REGISTERED

Table 4.2 Adding of User

Figure 4.5 Fingerprint Enrollment Sample #1

Figure 4.6 Fingerprint Enrollment Sample #2

Figure 4.7 Fingerprint Enrollment Sample #3

Based on the collected data for registering new fingerprint into the system, new fingerprints can be registered given that the system is in "Administrator Mode" where the testing results for the validating admin's fingerprint is shown in Table 4.1.

TRIAL	MODE	STATUS
1	SUCCESS	UNLOCKED
2	SUCCESS	UNLOCKED
3	SUCCESS	UNLOCKED
4	SUCCESS	UNLOCKED
5	SUCCESS	UNLOCKED
6	SUCCESS	UNLOCKED
7	SUCCESS	UNLOCKED
8	SUCCESS	UNLOCKED
9	SUCCESS	UNLOCKED
10	SUCCESS	UNLOCKED
11	SUCCESS	UNLOCKED
12	SUCCESS	UNLOCKED

13	SUCCESS	UNLOCKED
14	SUCCESS	UNLOCKED
15	SUCCESS	UNLOCKED
16	SUCCESS	UNLOCKED
17	SUCCESS	UNLOCKED
18	SUCCESS	UNLOCKED
19	SUCCESS	UNLOCKED
20	SUCCESS	UNLOCKED
21	SUCCESS	UNLOCKED
22	SUCCESS	UNLOCKED
23	SUCCESS	UNLOCKED
24	SUCCESS	UNLOCKED
25	SUCCESS	UNLOCKED
26	SUCCESS	UNLOCKED
27	SUCCESS	UNLOCKED
28	SUCCESS	UNLOCKED

29	SUCCESS	UNLOCKED
30	SUCCESS	UNLOCKED

Table 4.3 Validating User's Fingerprint

Figure 4.8 Firearm is Unlocked

Based on the gathered data, the system validates the user's fingerprint accurately resulting in unlocking the solenoid lock in the firearm's trigger. This proves that the design can secure firearms where it will be installed with 100% accuracy in verifying untrusted access.

TRIAL	NUMBER OF TRIGGER PULLED	STATUS
1	10	MESSAGE WAS SENT
2	20	MESSAGE WAS SENT
3	30	MESSAGE WAS SENT

4	40	MESSAGE WAS SENT
5	50	MESSAGE WAS SENT
6	60	MESSAGE WAS SENT
7	70	MESSAGE WAS SENT
8	80	MESSAGE WAS SENT
9	90	MESSAGE WAS SENT
10	100	MESSAGE WAS SENT

Table 4.4 Testing for monitoring of number of Shots/Time/Date

Based on the collected data, after pulling the trigger of the said firearms and locking it, a text message was sent to the registered phone number on the system having the number of times the trigger was pulled with its time and date. This proves that the design can monitor how many shots were made during its unlocked state.

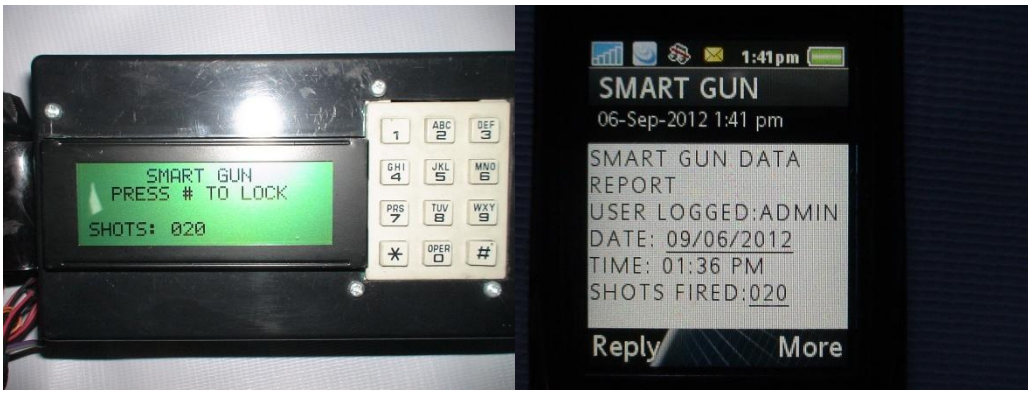


Figure 4.9 The total number of shots fired, date and time it was used.

Impact Analysis

The design would have a great effect on the safety of the people and stop the abuse and misuse of guns for crime. This will also prevent any other accidents that are gun-related. With this innovation, guns are more secured because only the rightful owner has access to this device with its built-in fingerprint scanner that would determine if the user owns the registered fingerprint or not. Also, passing and sharing of guns will not be possible because for every person, there is a unique finger print and only the registered owner has the access to pull the trigger. It can also affect the social relation of the agency to its clients ensuring the safety of their customers and having a more advanced weapon that is more secured than the conventional gun.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

Conclusion

In conclusion the Biometric analysis for firearm identification with real-time detection for ballistic projectile via SMS can be useful for keeping the safety of the gun users and avoiding the misuse of it. This paper achieved its objective to create a standalone device that is attached to a gun that asks first for the fingerprint of the registered user before the gun can actually be used. Also the gun lock would not be released unless a correct fingerprint pattern is used. With the use of the GSM technology, the device can send a message once an invalid fingerprint is used. Furthermore, the system keeps a record of the gun shots and sends them together with real time and date after the gun is disarmed. The design proves that the use of the fingerprint technology; can prevent any unwanted accidents in using firearms. It can also track the activity of the gun shots with the use of the GSM Module. Finally, the whole system would help in providing security in both the users and civilians.

Recommendation

Even though the design accomplished all the objectives that had been stated in this paper, there are still several limitations and improvements that can be addressed to make the design more functional.

This design proves that it can secure the firearms from unauthorized individuals but not 100% accurate. To make it use its full potential in terms of security, the researchers suggest that the solenoid lock is programmed to automatically lock, when its power source is less than 5%. When the switch is in unlocked state but its power source runs out, the device will remain unlocked. Furthermore, the researchers also suggest to have the said design sustain the heat and aftershock of every shot that is made by the gun.

Future studies relating to this paper can also include hand print scan, instead of just a finger print for the user. This can eliminate the time in which the user puts his/her finger on the scanner first, while having a hand print scan which can possibly attach to the grip of the firearm.

Bibliography

Jain, L.C. et al. (Eds.). 1999. Intelligent Biometric Techniques in Fingerprint and Face Recognition.

Langenburg, Glenn (January 24, 2005). Are one's fingerprints similar to those of his or her parents in any discernable way?".

Thornton, John (May 9, 2000). Latent Fingerprints, Setting Standards In The Comparison and Identification.

Diaz, Raul (2007). Biometrics: Security Vs Convenienc.

Meghdadi, Majid; Jalilzadeh, Saeed (29 October 2005). Validity and Acceptability of Results in Fingerprint Scanner.

Setlak, Dale. Advances in Biometric Fingerprint Technology are Driving Rapid Adoption in Consumer Marketplace.

Mazumdar, Subhra; Dhulipala, Venkata (2008). Biometric Security Using Finger Print Recognition.

Guodong Li, Hu Chen (2010). A New High-Level Security Potable System based on USB Key with Fingerprint

H.M.N. Dilum Bandara (2008). Biometrics

Younhee Gil, Dosung Ahn, Sungbum Pan, Yongwha Chung (2011). Access Control System with High Level Security Using Fingerprints

Zhaoxia Zhu, Fulong Chen (2011). Fingerprint Recognition-Based Access Controlling System for Automobiles

Anum Batool, Anam Tariq (2011). Computerized System for Fingerprint Identification for Biometric Security.

APPENDIX

A. OPERATOR'S MANUAL

1. SYSTEM REQUIREMENTS

The "Biometric analysis for firearm identification with real-time detection for ballistic projectile via SMS" only works if the Firearm locking system and the fingerprint reading device is turned on and the device can send messages if there is a load in it.

2. INSTALLATION PROCEDURE

The following procedures must be followed to ensure the system works properly.

Turning on the Firearm locking system:

1. Turn on the device.
2. Make sure the LCD displays no error.

Inserting the fingerprint reading device:

1. Make sure there is a battery connected into the device.
2. Push the # button to start up the device
3. Wait until the fingerprint scanner lights up.

Sending a message to the database

1. Make sure there is a load on the SIM card of the GSM module

2. Push the # button to start up the device
3. Wait until the fingerprint scanner lights up.
4. Place the fingerprint properly into the scanning glass
5. Finish the verification and check if the firearm unlocks
6. Pull the trigger
7. After pulling the trigger press the # button to lock it again and wait for the message to be received by the database.

3. USER'S MANUAL

3.1 Turn on the Device

3.2 Make sure the LCD displays no error.

3.3 Push the # button to start up the device.

3.4 Wait until the fingerprint scanner lights up.

3.5 Place the fingerprint properly into the scanning glass.

3.6 Finish the verification and check if the firearm unlocks.

4. TROUBLESHOOTING GUIDES AND PROCEDURES

4.1 After turning on the Biometric analysis for firearm identification with real-time detection for ballistic projectile via SMS, check if the LCD would display if the GSM Module has successfully initialized. If no, restart the system an wait for the GSM Module to initialize properly.

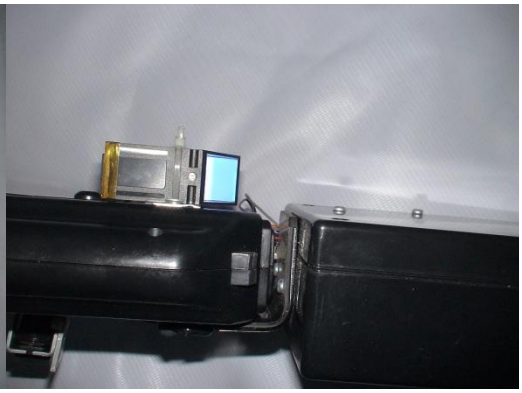
4.2 If the GSM Module fails to send the data reload the SIM card so that the GSM Module can send message again.

5. ERROR DEFINITIONS

5.1 GSM Module initializing error – the GSM Module fails to acquire network.

5.2 GSM Module error sending message – the system fails to send the message of the total number of shots and the data and time.

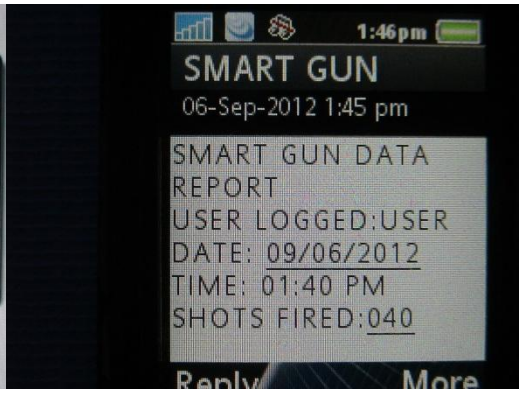
B. PICTURES OF PROTOTYPE



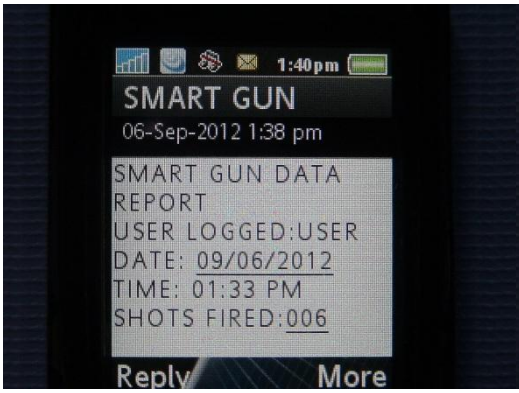
1.CHANGE FINGERPRINT
2.CHANGE SMS NUMBER
3.UNLOCK GUN
4.EXIT

1.CHANGE ADMIN FP
2.CHANGE USER FP
3.CANCEL

SMART GUN
PRESS # TO LOCK
SHOTS: 040


1:46pm
SMART GUN
06-Sep-2012 1:45 pm
SMART GUN DATA
REPORT
USER LOGGED:USER
DATE: 09/06/2012
TIME: 01:40 PM
SHOTS FIRED:040
Reply
More

SMART GUN
PRESS # TO UNLOCK


1:40pm
SMART GUN
06-Sep-2012 1:38 pm
SMART GUN DATA
REPORT
USER LOGGED:USER
DATE: 09/06/2012
TIME: 01:33 PM
SHOTS FIRED:006
Reply
More

1:41pm
SMART GUN
06-Sep-2012 1:41 pm
SMART GUN DATA
REPORT
USER LOGGED:ADMIN
DATE: 09/06/2012
TIME: 01:36 PM
SHOTS FIRED:020
Reply
More

C. PROGRAM LISTING

```
;-----------------------------------------------------------
; Code Produced by the Proton Compiler. Ver 3.5.2.5
; Copyright Rosetta Technologies/Crownhill Associates Ltd
; Written by Les Johnson. September 2011
;-----------------------------------------------------------
;
 LIST  P = 16F877A, F = INHX8M, W = 2, X = ON, R = DEC, MM = ON, N = 0, C = 255
INDF equ 0X0000
TMR0 equ 0X0001
PCL equ 0X0002
STATUS equ 0X0003
FSR equ 0X0004
PORTA equ 0X0005
PORTB equ 0X0006
PORTC equ 0X0007
PORTD equ 0X0008
PORTE equ 0X0009
PCLATH equ 0X000A
INTCON equ 0X000B
PIR1 equ 0X000C
PIR2 equ 0X000D
TMR1L equ 0X000E
TMR1LH equ 0X000F
TMR1H equ 0X000F
T1CON equ 0X0010
TMR2 equ 0X0011
T2CON equ 0X0012
SSPBUF equ 0X0013
SSPCON equ 0X0014
CCPR1L equ 0X0015
CCPR1LH equ 0X0016
CCPR1H equ 0X0016
CCP1CON equ 0X0017
RCSTA equ 0X0018
TXREG equ 0X0019
RCREG equ 0X001A
CCPR2L equ 0X001B
CCPR2LH equ 0X001C
CCPR2H equ 0X001C
CCP2CON equ 0X001D
ADRESLH equ 0X001E
ADRESH equ 0X001E
```

```
ADCON0 equ 0X001F
OPTION_REG equ 0X0081
TRISA equ 0X0085
TRISB equ 0X0086
TRISC equ 0X0087
TRISD equ 0X0088
TRISE equ 0X0089
PIE1 equ 0X008C
PIE2 equ 0X008D
PCON equ 0X008E
SSPCON2 equ 0X0091
PR2 equ 0X0092
SSPADD equ 0X0093
SSPSTAT equ 0X0094
TXSTA equ 0X0098
SPBRG equ 0X0099
CMCON equ 0X009C
CVRCON equ 0X009D
ADRESL equ 0X009E
ADCON1 equ 0X009F
EEDATL equ 0X010C
EEDATA equ 0X010C
EEDAT equ 0X010C
EEADR equ 0X010D
EEDATLH equ 0X010E
EEDATH equ 0X010E
EEADRH equ 0X010F
EECON1 equ 0X018C
EECON2 equ 0X018D
_I2C_SCL_PORT=TRISC
_I2C_SCL_PIN=3
_I2C_SDA_PORT=TRISC
_I2C_SDA_PIN=4
IRP=7
RP1=6
RP0=5
NOT_TO=4
NOT_PD=3
Z=2
DC=1
C=0
GIE=7
PEIE=6
T0IE=5
```

TMR0IE=5
INTE=4
RBIE=3
T0IF=2
TMR0IF=2
INTF=1
RBIF=0
PSPIF=7
ADIF=6
RCIF=5
PP_RCIF=5
TXIF=4
PP_TXIF=4
SSPIF=3
CCP1IF=2
TMR2IF=1
TMR1IF=0
CMIF=6
EEIF=4
BCLIF=3
CCP2IF=0
T1CKPS1=5
T1CKPS0=4
T1OSCEN=3
NOT_T1SYNC=2
T1INSYNC=2
T1SYNC=2
TMR1CS=1
TMR1ON=0
TOUTPS3=6
TOUTPS2=5
TOUTPS1=4
TOUTPS0=3
TMR2ON=2
PP_TMR2ON=2
T2CKPS1=1
PP_T2CKPS1=1
T2CKPS0=0
PP_T2CKPS0=0
WCOL=7
SSPOV=6
SSPEN=5
CKP=4
SSPM3=3

SSPM2=2
SSPM1=1
SSPM0=0
CCP1X=5
CCP1Y=4
CCP1M3=3
CCP1M2=2
CCP1M1=1
CCP1M0=0
SPEN=7
RX9=6
RC9=6
NOT_RC8=6
RC8_9=6
SREN=5
CREN=4
PP_CREN=4
ADDEN=3
FERR=2
OERR=1
PP_OERR=1
RX9D=0
RCD8=0
CCP2X=5
CCP2Y=4
CCP2M3=3
CCP2M2=2
CCP2M1=1
CCP2M0=0
ADCS1=7
ADCS0=6
CHS2=5
CHS1=4
CHS0=3
GO=2
NOT_DONE=2
GO_DONE=2
PP_GO_DONE=2
ADON=0
NOT_RBPU=7
INTEDG=6
T0CS=5
T0SE=4
PSA=3

PS2=2
PS1=1
PS0=0
IBF=7
OBF=6
IBOV=5
PSPMODE=4
TRISE2=2
TRISE1=1
TRISE0=0
PSPIE=7
ADIE=6
RCIE=5
TXIE=4
SSPIE=3
CCP1IE=2
TMR2IE=1
TMR1IE=0
CMIE=6
EEIE=4
BCLIE=3
CCP2IE=0
NOT_POR=1
NOT_BO=0
NOT_BOR=0
GCEN=7
ACKSTAT=6
PP_ACKSTAT=6
ACKDT=5
PP_ACKDT=5
ACKEN=4
PP_ACKEN=4
RCEN=3
PP_RCEN=3
PEN=2
PP_PEN=2
RSEN=1
PP_RSEN=1
SEN=0
PP_SEN=0
SMP=7
CKE=6
I2C_DATA=5
NOT_A=5

NOT_ADDRESS=5
D_A=5
DATA_ADDRESS=5
I2C_STOP=4
I2C_START=3
I2C_READ=2
NOT_W=2
NOT_WRITE=2
R_W=2
PP_R_W=2
READ_WRITE=2
UA=1
BF=0
PP_BF=0
CSRC=7
TX9=6
NOT_TX8=6
TX8_9=6
TXEN=5
SYNC=4
BRGH=2
TRMT=1
TX9D=0
TXD8=0
C2OUT=7
C1OUT=6
C2INV=5
C1INV=4
CIS=3
CM2=2
CM1=1
CM0=0
CVREN=7
CVROE=6
CVRR=5
CVR3=3
CVR2=2
CVR1=1
CVR0=0
ADFM=7
ADCS2=6
PCFG3=3
PCFG2=2
PCFG1=1

```
PCFG0=0
EEPGD=7
PP_EEPGD=7
WRERR=3
PP_WRERR=3
WREN=2
PP_WREN=2
WR=1
PP_WR=1
RD=0
PP_RD=0
  __MAXRAM 0X1FF
  __BADRAM 0X8F-0X90, 0X95-0X97, 0X9A-0X9B
  __BADRAM 0X105, 0X107-0X109
  __BADRAM 0X185, 0X187-0X189, 0X18E-0X18F
CP_ALL equ 0X1FFF
CP_OFF equ 0X3FFF
DEBUG_OFF equ 0X3FFF
DEBUG_ON equ 0X37FF
WRT_1FOURTH equ 0X3BFF
CPD_OFF equ 0X3FFF
CPD_ON equ 0X3EFF
LVP_ON equ 0X3FFF
LVP_OFF equ 0X3F7F
BODEN_ON equ 0X3FFF
BODEN_OFF equ 0X3FBF
PWRTE_OFF equ 0X3FFF
PWRTE_ON equ 0X3FF7
WDT_ON equ 0X3FFF
WDT_OFF equ 0X3FFB
RC_OSC equ 0X3FFF
HS_OSC equ 0X3FFE
XT_OSC equ 0X3FFD
LP_OSC equ 0X3FFC
BOREN_OFF equ 0X3FBF
BOREN_ON equ 0X3FFF
CP_ON equ 0X1FFF
FOSC_EXTRC equ 0X3FFF
FOSC_HS equ 0X3FFE
FOSC_LP equ 0X3FFC
FOSC_XT equ 0X3FFD
WDTE_OFF equ 0X3FFB
WDTE_ON equ 0X3FFF
WRT_1FOURTH equ 0X3BFF
```

```
WRT_256 equ 0X3DFF
WRT_HALF equ 0X39FF
WRT_OFF equ 0X3FFF
#define __16F877A 1
#define XTAL 20
#define _CORE 14
#define _MAXRAM 368
#define _RAM_END 0X0170
#define _MAXMEM 8192
#define _ADC 8
#define _ADC_RES 10
#define _EEPROM 256
#define _PAGES 4
#define _BANKS 3
#define RAM_BANKS 4
#define _USART 1
#define _USB 0
#define _FLASH 1
#define _CWRITE_BLOCK 8
#define BANK0_START 0X20
#define BANK0_END 0X7F
#define BANK1_START 0X00A0
#define BANK1_END 0X00EF
#define BANK2_START 0X0110
#define BANK2_END 0X016F
#define BANK3_START 0X0190
#define BANK3_END 0X01EF
#define _SYSTEM_VARIABLE_COUNT 40
ram_bank = 0
CURRENT@PAGE = 0
DEST@PAGE = 0
#define LCD#DTPORT PORTC
#define LCD#DTPIN 0
#define LCD#RSPORT PORTD
#define LCD#RSPIN 0
#define LCD#ENPORT PORTD
#define LCD#ENPIN 1
#define LCD#INTERFACE 4
#define LCD#LINES 4
#define LCD#TYPE 0
f@call macro PDEST
if (PDEST < 1)
if ((PDEST & 2048) == 0)
        bcf 10,3
```

```
else
        bsf 10,3
endif
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
else
if (PDEST > $)
if ((PDEST & 2048) == 0)
        bcf 10,3
else
        bsf 10,3
endif
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
else
if ((PDEST & 6144) == 0)
        clrf 10
else
if ((PDEST & 2048) == 0)
        bcf 10,3
else
        bsf 10,3
endif
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
endif
endif
endif
        call PDEST
        endm
F@JUMP macro PDEST
if (PDEST < 1)
if ((PDEST & 2048) == 0)
        bcf 10,3
else
```

```
        bsf 10,3
endif
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
else
if (PDEST > $)
if ((PDEST & 2048) == 0)
        bcf 10,3
else
        bsf 10,3
endif
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
else
if ((PDEST & 6144) == 0)
        clrf 10
else
if ((PDEST & 2048) == 0)
        bcf 10,3
else
        bsf 10,3
endif
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
endif
endif
endif
        goto PDEST
        endm
set@page macro PDEST
if ((PDEST & 2048) == 0)
        bcf 10,3
else
        bsf 10,3
endif
```

```
if ((PDEST & 4096) == 0)
        bcf 10,4
else
        bsf 10,4
endif
        endm
s@b     macro pVarin
if((pVarin & 384) == 0)
if(ram_bank == 1)
        bcf 3,5
endif
if(ram_bank == 2)
        bcf 3,6
endif
if(ram_bank == 3)
        bcf 3,5
        bcf 3,6
endif
ram_bank = 0
endif
if((pVarin & 384) == 128)
if(ram_bank == 0)
        bsf 3,5
endif
if(ram_bank == 2)
        bsf 3,5
        bcf 3,6
endif
if(ram_bank == 3)
        bcf 3,6
endif
ram_bank = 1
endif
if((pVarin & 384) == 256)
if(ram_bank == 0)
        bsf 3,6
endif
if(ram_bank == 1)
        bcf 3,5
        bsf 3,6
endif
if(ram_bank == 3)
        bcf 3,5
endif
```

```
ram_bank = 2
endif
if((pVarin & 384) == 384)
if(ram_bank == 0)
        bsf 3,5
        bsf 3,6
endif
if(ram_bank == 1)
        bsf 3,6
endif
if(ram_bank == 2)
        bsf 3,5
endif
ram_bank = 3
endif
        endm
r@b     macro
if((ram_bank & 1) != 0)
        bcf 3,5
endif
if((ram_bank & 2) != 0)
        bcf 3,6
endif
ram_bank = 0
        endm
jump macro PLABEL
    goto PLABEL
    endm
wreg_byte macro pByteOut
    s@b pByteOut
    movwf pByteOut
    r@b
    endm
wreg_bit macro pVarOut,pBitout
    s@b pVarOut
    andlw 1
    btfsc STATUS,2
    bcf pVarOut,pBitout
    btfss STATUS,2
    bsf pVarOut,pBitout
    r@b
    endm
wreg_word macro pWordOut
    s@b pWordOut
```

```
    movwf pWordOut
    s@b pWordOut+1
    clrf pWordOut+1
    r@b
    endm
wreg_dword macro pDwordOut
    s@b pDwordOut+3
    clrf pDwordOut+3
    s@b pDwordOut+2
    clrf pDwordOut+2
    s@b pDwordOut+1
    clrf pDwordOut+1
    s@b pDwordOut
    movwf pDwordOut
    r@b
    endm
byte_wreg macro pByteIn
    s@b pByteIn
    movf pByteIn,W
    r@b
    endm
num_wreg macro pNumIn
    movlw (pNumIn & 255)
    endm
num_byte macro pNumIn,pByteOut
    s@b pByteOut
if(pNumIn == 0)
    clrf pByteOut
else
    movlw (pNumIn & 255)
    movwf pByteOut
endif
    r@b
    endm
num_bit macro pNumIn,pVarOut,pBitout
    s@b pVarOut
if((pNumIn & 1) == 1)
    bsf pVarOut,pBitout
else
    bcf pVarOut,pBitout
endif
    r@b
    endm
num_word macro pNumIn,pWordOut
```

```
if((pNumIn & 255) == 0)
   s@b pWordOut
   clrf pWordOut
else
   s@b pWordOut
   movlw low (pNumIn)
   movwf pWordOut
endif
if(((pNumIn >> 8) & 255) == 0)
   s@b pWordOut+1
   clrf pWordOut+1
else
   s@b pWordOut+1
   movlw high (pNumIn)
   movwf pWordOut+1
endif
   r@b
   endm
num_dword macro pNumIn,pDwordOut
if ((pNumIn >> 24 & 255) == 0)
   s@b pDwordOut+3
   clrf pDwordOut+3
else
   s@b pDwordOut+3
   movlw ((pNumIn >> 24) & 255)
   movwf pDwordOut+3
endif
if( ((pNumIn >> 16) & 255) == 0)
   s@b pDwordOut+2
   clrf pDwordOut+2
else
   s@b pDwordOut+2
   movlw ((pNumIn >> 16) & 255)
   movwf pDwordOut+2
endif
if( ((pNumIn >> 8) & 255) == 0)
   s@b pDwordOut+1
   clrf pDwordOut+1
else
   s@b pDwordOut+1
   movlw high (pNumIn)
   movwf pDwordOut+1
endif
if((pNumIn & 255) == 0)
```

```
    s@b pDwordOut
    clrf pDwordOut
else
    s@b pDwordOut
    movlw low (pNumIn)
    movwf pDwordOut
endif
    r@b
    endm
bit_wreg macro pVarin,pBitIn
    s@b pVarin
    clrw
    btfsc pVarin,pBitIn
    movlw 1
    r@b
    endm
bit_byte macro pVarin,pBitIn,pByteOut
    s@b pVarin
    clrw
    btfsc pVarin,pBitIn
    movlw 1
    s@b pByteOut
    movwf pByteOut
    r@b
    endm
bit_bit macro pVarin,pBitIn,pVarOut,pBitout
if((pVarin & 65408) == (pVarOut & 65408))
    s@b pVarOut
    btfsc pVarin,pBitIn
    bsf pVarOut,pBitout
    btfss pVarin,pBitIn
    bcf pVarOut,pBitout
else
    s@b pVarin
    clrdc
    btfsc pVarin,pBitIn
    setdc
    s@b pVarOut
    skpndc
    bsf pVarOut,pBitout
    skpdc
    bcf pVarOut,pBitout
endif
    endm
```

```
bit_word macro pVarin,pBitIn,pWordOut
   s@b pWordOut+1
   clrf pWordOut+1
   bit_byte pVarin,pBitIn,pWordOut
   endm
bit_dword macro pVarin,pBitIn,pDwordOut
   s@b pDwordOut+3
   clrf pDwordOut+3
   s@b pDwordOut+2
   clrf pDwordOut+2
   s@b pDwordOut+1
   clrf pDwordOut+1
   bit_byte pVarin,pBitIn,pDwordOut
   endm
word_wreg macro pWordIn
   byte_wreg pWordIn
   endm
word_byte macro pWordIn,pByteOut
   byte_byte pWordIn,pByteOut
   endm
word_bit macro pWordIn,pVarOut,pBitout
   byte_bit pWordIn, pVarOut, pBitout
   endm
word_word macro pWordIn,pWordOut
   s@b pWordIn+1
   movf pWordIn+1,W
   s@b pWordOut+1
   movwf pWordOut+1
   byte_byte pWordIn,pWordOut
   endm
word_dword macro pWordIn,pDwordOut
   s@b pDwordOut+3
   clrf pDwordOut+3
   s@b pDwordOut+2
   clrf pDwordOut+2
   byte_byte pWordIn+1,pDwordOut+1
   byte_byte pWordIn,pDwordOut
   endm
byte_byte macro pByteIn,pByteOut
   s@b pByteIn
   movf pByteIn,W
   s@b pByteOut
   movwf pByteOut
   r@b
```

```
    endm
byte_word macro pByteIn,pWordOut
    s@b pWordOut+1
    clrf pWordOut+1
    byte_byte pByteIn,pWordOut
    endm
byte_dword macro pByteIn,pDwordOut
    s@b pDwordOut+3
    clrf pDwordOut+3
    s@b pDwordOut+2
    clrf pDwordOut+2
    s@b pDwordOut+1
    clrf pDwordOut+1
    byte_byte pByteIn,pDwordOut
    endm
    byte_bit macro pByteIn,pVarOut,pBitout
if((pByteIn & 65408) == (pVarOut & 65408))
    s@b pByteIn
    btfsc pByteIn,0
    bsf pVarOut,pBitout
    btfss pByteIn,0
    bcf pVarOut,pBitout
else
    s@b pByteIn
    rrf pByteIn,W
    s@b pVarOut
    skpnc
    bsf pVarOut,pBitout
    skpc
    bcf pVarOut,pBitout
endif
    r@b
    endm
dword_wreg macro pDwordIn
    byte_wreg pDwordIn
    endm
dword_byte macro pDwordIn,pByteOut
    byte_byte pDwordIn,pByteOut
    endm
dword_word macro pDwordIn,pWordOut
    s@b pDwordIn+1
    movf pDwordIn+1,W
    s@b pWordOut+1
    movwf pWordOut+1
```

```
    byte_byte pDwordIn,pWordOut
    endm
dword_dword macro pDwordIn,pDwordOut
    byte_byte pDwordIn+3,pDwordOut+3
    byte_byte pDwordIn+2,pDwordOut+2
    byte_byte pDwordIn+1,pDwordOut+1
    byte_byte pDwordIn,pDwordOut
    endm
dword_bit macro pDwordIn,pVarOut,pBitout
    byte_bit pDwordIn,pVarOut,pBitout
    endm
variable CURRENT@PAGE = 0
variable PDESTINATION@PAGE = 0
FIND@PAGE macro pLabelIn
local CURRENT_ADDR = $
local DEST_ADDR = pLabelIn
if((CURRENT_ADDR >= 0X1800) && (CURRENT_ADDR <= 0X2000))
    CURRENT@PAGE = 3
endif
if((CURRENT_ADDR >= 0X1000) && (CURRENT_ADDR <= 0X1800))
    CURRENT@PAGE = 2
endif
if((CURRENT_ADDR >= 0X0800) && (CURRENT_ADDR <= 0X1000))
    CURRENT@PAGE = 1
endif
if((CURRENT_ADDR >= 0) && (CURRENT_ADDR <= 0X0800))
    CURRENT@PAGE = 0
endif
if((DEST_ADDR >= 0X1800) && (DEST_ADDR <= 0X2000))
    PDESTINATION@PAGE = 3
endif
if((DEST_ADDR >= 0X1000) && (DEST_ADDR <= 0X1800))
    PDESTINATION@PAGE = 2
endif
if((DEST_ADDR >= 0X0800) && (DEST_ADDR <= 0X1000))
    PDESTINATION@PAGE = 1
endif
if((DEST_ADDR >= 0) && (DEST_ADDR <= 0X0800))
    PDESTINATION@PAGE = 0
endif
    endm
NUM_FSR macro pNumIn
    num_byte pNumIn, FSR
if (((pNumIn >> 8) & 255) == 1)
```

```
    bsf STATUS,7
else
    bcf STATUS,7
endif
    endm
label_word macro pLabelIn,pWordOut
    movlw low (pLabelIn)
    s@b pWordOut
    movwf pWordOut
    movlw high (pLabelIn)
    s@b pWordOut
    movwf pWordOut+1
    r@b
    endm
BPF = 32
BPFH = 33
CSTR#CT = 34
GEN = 35
GEN2 = 36
GEN2H = 37
GEN3 = 38
GEN3H = 39
GEN4 = 40
GEN4H = 41
GENH = 42
GPR = 43
PBP#VAR0 = 44
PBP#VAR1 = 45
PBP#VAR2 = 46
PP0 = 47
PP0H = 48
PP0HH = 49
PP0HHH = 50
PP1 = 51
PP1H = 52
PP2 = 53
PP2H = 54
PP2HH = 55
PP2HHH = 56
PP3 = 57
PP3H = 58
PP4 = 59
PP5 = 60
PP5H = 61
```

PP6 = 62
PP6H = 63
PP7 = 64
PP7H = 65
PPZ = 66
PPZH = 67
PPZHH = 68
PPZHHH = 69
RPFH = 70
SP#P9 = 71
FPC = 72
FPECode = 73
FPEMsg = 74
variable FPEMsg#0=74,FPEMsg#1=75,FPEMsg#2=76,FPEMsg#3=77
variable FPEMsg#4=78,FPEMsg#5=79,FPEMsg#6=80,FPEMsg#7=81
variable FPEMsg#8=82,FPEMsg#9=83,FPEMsg#10=84,FPEMsg#11=85
variable FPEMsg#12=86,FPEMsg#13=87,FPEMsg#14=88,FPEMsg#15=89
variable FPEMsg#16=90,FPEMsg#17=91,FPEMsg#18=92,FPEMsg#19=93
variable FPEMsg#20=94
sData = 95
variable sData#0=95,sData#1=96,sData#2=97,sData#3=98
variable sData#4=99,sData#5=100,sData#6=101,sData#7=102
variable sData#8=103,sData#9=104,sData#10=105,sData#11=106
variable sData#12=107,sData#13=108,sData#14=109,sData#15=110
variable sData#16=111,sData#17=112,sData#18=113,sData#19=114
variable sData#20=115,sData#21=116,sData#22=117,sData#23=118
sReply = 119
variable sReply#0=119,sReply#1=120,sReply#2=121,sReply#3=122
variable sReply#4=123,sReply#5=124,sReply#6=125,sReply#7=126
variable sReply#8=127,sReply#9=160,sReply#10=161,sReply#11=162
variable sReply#12=163,sReply#13=164,sReply#14=165,sReply#15=166
variable sReply#16=167,sReply#17=168,sReply#18=169,sReply#19=170
variable sReply#20=171,sReply#21=172,sReply#22=173,sReply#23=174
chkID = 175
chkTmp = 176
chkTmpH = 177
userID = 178
userIDH = 179
kpCtr = 180
keytemp = 181
gCtr = 182
logRetry = 183
shotCtr = 184
uNum1 = 185

variable uNum1#0=185,uNum1#1=186,uNum1#2=187,uNum1#3=188
variable uNum1#4=189,uNum1#5=190,uNum1#6=191,uNum1#7=192
variable uNum1#8=193,uNum1#9=194,uNum1#10=195,uNum1#11=196
timeStamps = 197
variable timeStamps#0=197,timeStamps#1=198,timeStamps#2=199,timeStamps#3=200
variable timeStamps#4=201,timeStamps#5=202,timeStamps#6=203,timeStamps#7=204
variable
timeStamps#8=205,timeStamps#9=206,timeStamps#10=207,timeStamps#11=208
variable
timeStamps#12=209,timeStamps#13=210,timeStamps#14=211,timeStamps#15=212
variable
timeStamps#16=213,timeStamps#17=214,timeStamps#18=215,timeStamps#19=216
variable timeStamps#20=217
tmpVal = 218
variable tmpVal#0=218,tmpVal#1=219,tmpVal#2=220
years = 221
yearsH = 222
months = 223
days = 224
hr = 225
mn = 226
sc = 227
ampm = 228
#define __LCD_DTPORT PORTC
#define __LCD_RSPORT PORTD
#define __LCD_ENPORT PORTD
#define kpRow1 PORTB,0
#define kpRow2 PORTB,1
#define kpRow3 PORTB,2
#define kpRow4 PORTB,3
#define kpCol1 PORTB,4
#define kpCol2 PORTB,5
#define kpCol3 PORTB,6
#define gUnlock PORTD,2
#define gLock PORTD,3
#define myBuzzer PORTC,4
#define gsmPW PORTC,5
#define trigTS PORTD,6
#define chkSLow chkTmp
#define chkSHi chkTmpH
#define uIDL userID
#define uIDH userIDH
#define __XTAL 20
#define __HSERIAL_BAUD 9600

```
#define __HSERIAL_RCSTA 144
#define __HSERIAL_TXSTA 36
#define __LCD_DTPIN 0
#define __LCD_RSPIN 0
#define __LCD_ENPIN 1
#define __LCD_LINES 4
#define __LCD_INTERFACE 4
HSERIAL_SPBRG = 129
HSERIAL_TXSTA = 36
HSERIAL_RCSTA = 144
HSERIAL_BAUD = 9600
proton#code#start
        org 0
        nop
        movlw high (proton#main#start)
        movwf 10
        goto proton#main#start
        org 4
FIND@ADDRESS
        bcf 3,7
        movwf 62
        clrf 63
        rlf 63,F
        movlw 0
        subwf 63,W
        movlw 128
        skpnz
        subwf 62,W
        skpc
        goto $ + 7
        btfsc 58,7
        goto $ + 5
        movlw 32
        addwf 62,F
        skpnc
        incf 63,F
        movlw 0
        subwf 63,W
        movlw 240
        skpnz
        subwf 62,W
        skpc
        goto $ + 6
        bsf 3,7
```

```
        movlw 32
        addwf 62,F
        skpnc
        incf 63,F
        movlw 1
        subwf 63,W
        movlw 112
        skpnz
        subwf 62,W
        skpc
        goto FA@EXT
        movlw 32
        addwf 62,F
FA@EXT
        movf 62,W
        movwf 4
        return
LCD@CLS
        movlw 128
        movwf 33
CLS
        movlw 254
        call CH@SND
        movlw 1
        call CH@SND
        movlw 117
        movwf 48
        movlw 48
        goto DLUS@W
LCD@CUR
        clrf 33
        bsf 33,7
CURSOR
        movwf 60
        tstf 60
        skpz
        decf 60,F
        movlw 254
        call CH@SND
        tstf 35
        skpnz
        goto $ + 3
        decfsz 35,W
        goto $ + 3
```

```
        movlw 128
        goto $ + 18
        movf 35,W
        xorlw 2
        skpz
        goto $ + 3
        movlw 192
        goto $ + 12
        movf 35,W
        xorlw 3
        skpz
        goto $ + 3
        movlw 148
        goto $ + 6
        movf 35,W
        xorlw 4
        skpz
        goto $ + 3
        movlw 212
        movwf 35
        movf 60,W
        addwf 35,W
        goto CH@SND
LCD@CRS
        clrf 33
        bsf 33,7
CURS@
        movwf 60
        movlw 254
        call CH@SND
        movf 60,W
        goto CH@SND
IN@DEC32
        clrf 53
        clrf 54
        clrf 55
        clrf 56
        bcf 43,5
IN@DECB32
        call CH@RCV
        skpc
        return
        addlw 211
        skpnz
```

```
        bsf 43,5
        addlw 45
        addlw 198
        skpnc
        goto IN@DECB32
        addlw 10
        skpc
        goto IN@DECB32
IN@DCL132
        movwf 51
        clrf 50
        clrf 49
        clrf 48
        movlw 10
        movwf 47
        call AL@MUL
        movf 51,W
        addwf 47,W
        movwf 53
        movf 48,W
        movwf 54
        clrw
        skpnc
        movlw 1
        addwf 54,F
        movf 49,W
        movwf 55
        clrw
        skpnc
        movlw 1
        addwf 55,F
        movf 50,W
        movwf 56
        clrw
        skpnc
        movlw 1
        addwf 56,F
        decf 41,F
        skpnz
        goto IN@BNF32
        call CH@RCV
        skpc
        return
        addlw 198
```

```
        skpnc
        goto IN@BNF32
        addlw 10
        skpnc
        goto IN@DCL132
IN@BNF32
        btfss 43,5
        goto $ + 12
        comf 53,F
        comf 54,F
        comf 55,F
        comf 56,F
        incf 53,F
        skpnz
        incf 54,F
        skpnz
        incf 55,F
        skpnz
        incf 56,F
        movf 53,W
        setc
        goto I@NT
OUT@DECB
        clrf 41
OUT@DECC
        movwf 53
        clrf 54
OUT@DEC
        bcf 32,3
        movf 41,W
        skpnz
        bsf 32,3
        movlw 5
        movwf 40
        movlw 39
        movwf 52
        movlw 16
        call D@DIG
        movlw 3
        movwf 52
        movlw 232
        call D@DIG
        clrf 52
        movlw 100
```

```
        call D@DIG
        clrf 52
        movlw 10
        call D@DIG
        movf 53,W
        goto SEND@IT
D@DIG
        movwf 51
        movf 54,W
        movwf 48
        movf 53,W
        movwf 47
        call D@VD
        movf 47,W
SEND@IT
        movwf 47
        decf 40,F
        skpnz
        bcf 32,3
        movf 41,W
        skpnz
        goto $ + 4
        subwf 40,W
        skpnc
        goto EX@SEND@IT
        movf 47,W
        skpz
        bcf 32,3
        btfsc 32,3
        goto EX@SEND@IT
        addlw 48
        goto CH@SND
EX@SEND@IT
        return
SB@STROUT
        movwf 53
SB@STROUTLP
        movf 40,W
        skpnz
        goto I@NT
        btfsc 54,0
        bsf     3,7
        movf 53,W
        movwf 4
```

```
        movf 0,W
        bcf     3,7
        skpnz
        goto I@NT
        call CH@SND
        incf 53,F
        decf 40,F
        goto SB@STROUTLP
PRINT
        movwf 58
        bcf PORTD,1
        bcf PORTD,0
        bsf 3,5
        bcf PORTD,1
        bcf PORTD,0
        movlw 240
        andwf PORTC,F
        bcf 3,5
        movf 58,W
        btfsc 32,1
        goto PRT@1
        movlw 62
        movwf 48
        movlw 128
        call DLUS@W
        movlw 51
        movwf 57
        call PR@LP
        movlw 19
        movwf 48
        movlw 136
        call DLUS@W
        call PR@LP
        movlw 100
        call DL@US
        call PR@LP
        movlw 100
        call DL@US
        movlw 34
        movwf 57
        call PR@LP
        movlw 40
        call PRT@C
        movlw 12
```

```
        call PRT@C
        movlw 6
        call PRT@C
        bsf 32,1
        movf 58,W
        goto PRT@1
PRT@C
        bsf 32,0
PRT@1
        movwf 57
        btfss 32,0
        goto PRIT@SKP
        bcf PORTD,0
        sublw 3
        skpc
        goto PRT@S
        call PRT@S
        movlw 7
        movwf 48
        movlw 208
        call DLUS@W
        bsf 3,0
        return
PRIT@SKP
        bsf 32,0
        sublw 254
        skpnz
        goto PR@EX
        bsf PORTD,0
PRT@S
        swapf 57,F
        btfss 32,0
PR@LP
        bcf 32,0
        bsf PORTD,1
        movlw 240
        andwf PORTC,F
        movf 57,W
        andlw 15
        iorwf PORTC,F
        nop
        bcf PORTD,1
        swapf 57,F
        nop
```

```
        btfsc 32,0
        goto PR@LP
        movlw 50
        call DL@US
PR@EX
        movf 58,W
        goto I@NT
SERIN
        call SRIN@B
        skpnc
        goto SERIN
SRIN@S
        call SRIN@SD
        clrf 59
        movlw 8
        movwf 52
        call SRIN@D
        call SRIN@B
        skpnc
        incf 59,F
        btfss 61,5
        clrf 59
        rrf 51,F
        decfsz 52,F
        goto $ - 8
        btfsc 61,5
        bcf 51,7
        call SRIN@D
        movf 51,W
        return
SRIN@B
        movf 35,W
        movwf 4
        movf 42,W
        bsf 4,7
        iorwf 0,F
        bcf 4,7
        andwf 0,W
        btfsc 61,6
        xorwf 42,W
        addlw 255
        return
SEROUT
        movwf 57
```

```
SRO@X
        movf 35,W
        movwf 4
        clrf 43
        movlw 240
        andwf 43,F
        movlw 8
        movwf 58
        bcf 3,0
        skpnc
        incf 43,F
        call SRO@B
        rrf 57,F
        decfsz 58,F
        goto $ - 5
        nop
        btfsc 61,5
        rrf 43,W
        call SRO@B
        goto $ + 1
        bsf 3,0
        call SRO@B
        movf 39,W
        movwf 52
        movf 38,W
        call DLY@W
        goto I@NT
SRO@B
        bcf 4,7
        btfss 61,7
        goto $ + 13
        movf 0,W
        iorwf 42,W
        btfss 61,6
        xorwf 42,W
        movwf 0
        bsf 4,7
        movf 0,W
        iorwf 42,W
        skpc
        xorwf 42,W
        movwf 0
        goto SRIN@D
        movf 0,W
```

```
        iorwf 42,W
        skpc
        xorwf 42,W
        btfsc 61,6
        xorwf 42,W
        movwf 0
        bsf 4,7
        comf 42,W
        andwf 0,F
        goto SRIN@D
SRIN@SD
        bsf 43,6
SRIN@D
        movf 61,W
        andlw 31
        addlw 0
        movwf 48
        movf 60,W
        addlw 14
        skpnc
        incf 48,F
        btfss 43,6
        goto DLUS@W
        bcf 43,6
        movwf 47
        movlw 2
        call R@SH
        goto DLUS@W
HRSIN2
        btfsc 24,PP_OERR
        bcf 24,PP_CREN
        bsf 24,PP_CREN
        movf 35,W
        movwf 47
        movf 42,W
        movwf 48
        clrf 51
        clrf 52
        goto $ + 1
        goto $ + 1
HRSIN@LPT
        nop
        btfsc 12,PP_RCIF
        goto H@RSIN
```

```
        movlw 255
        addwf 51,F
        skpc
        addwf 52,F
        skpc
        addwf 47,F
        skpc
        addwf 48,F
        skpc
        goto I@NT
        incfsz 51,W
        goto $ - 16
        incfsz 52,W
        goto $ - 17
        movlw 250
        movwf 51
        clrf 52
        goto HRSIN@LPT
HRSIN
        btfsc 24,PP_OERR
        bcf 24,PP_CREN
        bsf 24,PP_CREN
        btfss 12,PP_RCIF
        goto $ - 1
H@RSIN
        movf 26,W
        bsf 3,0
        goto I@NT
HRSOUT
        btfss 12,PP_TXIF
        goto $ - 1
        movwf 25
        bsf 3,0
        goto I@NT
CH@SND
        btfsc 33,7
        goto PRINT
        btfsc 33,5
        goto SEROUT
        btfsc 33,2
        goto HRSOUT
        return
CH@RCV
        movf 70,F
```

```
        skpnz
        goto HRSIN2
        decf 70,W
        skpnz
        goto HRSIN
        movf 70,W
        xorlw 5
        skpnz
        goto SERIN
        bsf 3,0
        btfss 70,7
        return
        movf 0,W
        incf 4,F
        return
C@STROUT
        movwf 40
C@STROUTW
        clrf 34
C@STROUTLP
        movf 40,W
        bsf 3,6
        movwf 269
        bcf 3,6
        movf 41,W
        bsf 3,6
        movwf 271
        call C@READ
        skpnz
        goto I@NT
        call CH@SND
        incf 40,F
        skpnz
        incf 41,F
        incfsz 34,F
        goto C@STROUTLP
        goto I@NT
C@READ
        bsf 3,5
        bsf 3,6
        bsf 396,PP_EEPGD
        bsf 396,PP_RD
        nop
        nop
```

```
        bcf 3,5
        movf 268,W
        goto I@NT
DL@MS
        clrf 52
DLY@W
        movwf 51
DLY@P
        movlw 255
        addwf 51,F
        skpc
        addwf 52,F
        skpc
        goto I@NT
        movlw 3
        movwf 48
        movlw 230
        call DLUS@W
        goto DLY@P
DL@US
        clrf 48
DLUS@W
        addlw 252
        movwf 47
        comf 48,F
        skpc
        goto $ + 7
        movlw 255
        nop
        addwf 47,F
        skpnc
        goto $ - 3
        addwf 47,F
        nop
        incfsz 48,F
        goto $ - 8
        return
L@KL
        iorlw 0
        skpnz
        goto $ + 22
        movwf 59
        movf 47,W
        movwf 4
```

```
        btfsc 48,0
        bsf 3,7
        clrf 48
        clrf 47
        btfsc 59,7
        goto $ + 9
        movf 0,W
        movwf 47
        btfss 59,1
        goto $ + 9
        incf 4,F
        movf 0,W
        movwf 48
        goto $ + 5
        call C@BT
        andwf 0,W
        skpz
        incf 47,F
        goto I@NT
C@BT
        movlw (C@TBL >> 8)
        movwf 10
        movf 59,W
        andlw 7
        addwf 2,F
C@TBL
        retlw 1
        retlw 2
        retlw 4
        retlw 8
        retlw 16
        retlw 32
        retlw 64
        retlw 128
D@VD
        clrf 54
        clrf 53
D@VD2
        movlw 16
        movwf 57
        rlf 48,W
        rlf 53,F
        rlf 54,F
        movf 51,W
```

```
        subwf 53,F
        movf 52,W
        skpc
        incfsz 52,W
        subwf 54,F
        skpnc
        goto $ + 8
        movf 51,W
        addwf 53,F
        movf 52,W
        skpnc
        incfsz 52,W
        addwf 54,F
        bcf 3,0
        rlf 47,F
        rlf 48,F
        decfsz 57,F
        goto $ - 21
        movf 47,W
        return
AL@MUL
        movlw 32
        movwf 57
        clrf 66
        clrf 67
        clrf 68
        clrf 69
MLT32@LP
        clrc
        btfss 47,0
        goto MLT32@BR
        movf 53,W
        addwf 66,F
        movf 54,W
        skpnc
        incfsz 54,W
        addwf 67,F
        movf 55,W
        skpnc
        incfsz 55,W
        addwf 68,F
        movf 56,W
        skpnc
        incfsz 56,W
```

```
        addwf 69,F
MLT32@BR
        rrf 69,F
        rrf 68,F
        rrf 67,F
        rrf 66,F
        rrf 50,F
        rrf 49,F
        rrf 48,F
        rrf 47,F
        decfsz 57,F
        goto MLT32@LP
        return
        bcf 3,0
        rrf 48,F
        rrf 47,F
R@SH
        addlw 255
        skpnc
        goto $ - 5
        movf 47,W
        goto I@NT
I@NT
        bcf 3,7
I@NT2
        bcf 3,5
        bcf 3,6
        return
PUT@ARRAY
        movwf 57
PUT@ARRAYW
        movf 40,W
        movwf 58
        addwf 41,W
        call FIND@ADDRESS
        movf 57,W
        movwf 0
        goto I@NT
GET@ARRAY
        movwf 57
GET@ARRAYW
        movf 39,W
        movwf 58
        movf 57,W
```

```
        addwf 39,W
        call FIND@ADDRESS
        movf 0,W
        goto I@NT
VALDEC@32
        movwf 4
        clrf 41
        movlw 128
        movwf 70
        goto IN@DEC32
proton#main#start
        clrf 3
        bsf 3,5
        movlw 129
        movwf 153
        movlw 36
        movwf 152
        bcf 3,5
        movlw 144
        movwf 24
        clrf 32
F2_SOF equ $ ; MITFPGUN.PRP
F2_EOF equ $ ; MITFPGUN.PRP
F1_SOF equ $ ; MITFPGUN.BAS
F1_000006 equ $ ; IN [MITFPGUN.BAS] DECLARE ALL_DIGITAL = TRUE
        movlw 7
        bsf STATUS,5
ram_bank = 1
        movwf CMCON
        movwf ADCON1
F1_000033 equ $ ; IN [MITFPGUN.BAS] TRISA = $FF
        movlw 255
        movwf TRISA
F1_000034 equ $ ; IN [MITFPGUN.BAS] TRISB = $0F
        movlw 15
        movwf TRISB
F1_000035 equ $ ; IN [MITFPGUN.BAS] TRISC = $90
        movlw 144
        movwf TRISC
F1_000036 equ $ ; IN [MITFPGUN.BAS] TRISD = $E0
        movlw 224
        movwf TRISD
F1_000037 equ $ ; IN [MITFPGUN.BAS] TRISE = $07
        movlw 7
```

```
        movwf TRISE
F1_000068 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        bcf STATUS,5
ram_bank = 0
        f@call DL@MS
F1_000071 equ $ ; IN [MITFPGUN.BAS] KPCOL1 = 0
        bcf PORTB,4
F1_000072 equ $ ; IN [MITFPGUN.BAS] KPCOL2 = 1
        bsf PORTB,5
F1_000073 equ $ ; IN [MITFPGUN.BAS] KPCOL3 = 0
        bcf PORTB,6
F1_000075 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_LOCKGUN
        f@call subs_LockGun
F1_000077 equ $ ; IN [MITFPGUN.BAS] IF KPROW4 = 0 THEN
        set@page BC@LL2
        btfsc PORTB,3
        goto BC@LL2
F1_000079 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_HWDIAGNOSTICS
        f@call subs_HWDiagnostics
F1_000081 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL2
F1_000083 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_SMS_INIT
        f@call subs_SMS_Init
mainCode
F1_000087 equ $ ; IN [MITFPGUN.BAS] LOGRETRY = 0
        bsf STATUS,5
ram_bank = 1
        clrf logRetry
jmp_mcStart
        bcf STATUS,5
ram_bank = 0
F1_000091 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_LOCKGUN
        f@call subs_LockGun
F1_000093 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000094 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"     SMART GUN"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB1)
        movwf GEN4H
        movlw low (STR@LB1)
        f@call C@STROUT
```

```
F1_000095 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1," PRESS # TO UNLOCK"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB2)
        movwf GEN4H
        movlw low (STR@LB2)
        f@call C@STROUT
F1_000097 equ $ ; IN [MITFPGUN.BAS] IF LOGRETRY > 0 THEN
        bsf STATUS,5
ram_bank = 1
        movf logRetry,F
        set@page BC@LL4
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,2
        goto BC@LL4
F1_000099 equ $ ; IN [MITFPGUN.BAS] PRINT AT 4,1,"LOGIN ATTEMPTS: ", DEC
LOGRETRY, "/3"
        movlw 128
        movwf BPFH
        movlw 212
        f@call LCD@CRS
        movlw high (STR@LB3)
        movwf GEN4H
        movlw low (STR@LB3)
        f@call C@STROUT
        bsf STATUS,5
ram_bank = 1
        movf logRetry,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECB
        movlw 47
        f@call PRINT
        movlw 51
        f@call PRINT
F1_000101 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL4
F1_000103 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL5
F1_000105 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
```

```
ram_bank = 1
        clrf keytemp
F1_000106 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000108 equ $ ; IN [MITFPGUN.BAS] IF KEYTEMP = "#" THEN
        movlw 35
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL8
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL8
F1_000110 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1, "VERIFYING FPRINT    "
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB4)
        movwf GEN4H
        movlw low (STR@LB4)
        f@call C@STROUT
F1_000112 equ $ ; IN [MITFPGUN.BAS] FPC = 0
        clrf FPC
F1_000113 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCOMMAND
        f@call subs_FPCommand
F1_000115 equ $ ; IN [MITFPGUN.BAS] SELECT CASE USERID
F1_000117 equ $ ; IN [MITFPGUN.BAS] CASE 0
        bsf STATUS,5
ram_bank = 1
        movf userIDH,W
        iorwf userID,W
        set@page BC@LL10
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL10
F1_000119 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1, "ACCESS DENIED      "
        movlw 128
        movwf BPFH
        movlw 148
```

```
        f@call LCD@CRS
        movlw high (STR@LB5)
        movwf GEN4H
        movlw low (STR@LB5)
        f@call C@STROUT
F1_000121 equ $ ; IN [MITFPGUN.BAS] IF LOGRETRY < 3 THEN
        movlw 3
        bsf STATUS,5
ram_bank = 1
        subwf logRetry,W
        set@page BC@LL13
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto BC@LL13
F1_000123 equ $ ; IN [MITFPGUN.BAS] INC LOGRETRY
        bsf STATUS,5
ram_bank = 1
        incf logRetry,F
F1_000124 equ $ ; IN [MITFPGUN.BAS] DELAYMS 2000
        movlw 7
        bcf STATUS,5
ram_bank = 0
        movwf PP1H
        movlw 208
        f@call DLY@W
F1_000125 equ $ ; IN [MITFPGUN.BAS] GOTO JMP_MCSTART
        F@JUMP jmp_mcStart
        F@JUMP BC@LL14
BC@LL13
F1_000127 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000129 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL14
        F@JUMP BC@LL9
BC@LL10
F1_000131 equ $ ; IN [MITFPGUN.BAS] CASE 1
        bsf STATUS,5
ram_bank = 1
        decf userID,W
        iorwf userIDH,W
        set@page BC@LL16
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
```

```
        goto BC@LL16
jmp_AdminStart
F1_000137 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000138 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"1.CHANGE FINGERPRINT"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB6)
        movwf GEN4H
        movlw low (STR@LB6)
        f@call C@STROUT
F1_000139 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"2.CHANGE SMS NUMBER"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB7)
        movwf GEN4H
        movlw low (STR@LB7)
        f@call C@STROUT
F1_000140 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"3.UNLOCK GUN"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB8)
        movwf GEN4H
        movlw low (STR@LB8)
        f@call C@STROUT
F1_000141 equ $ ; IN [MITFPGUN.BAS] PRINT AT 4,1,"4.EXIT"
        movlw 128
        movwf BPFH
        movlw 212
        f@call LCD@CRS
        movlw high (STR@LB9)
        movwf GEN4H
        movlw low (STR@LB9)
        f@call C@STROUT
F1_000143 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL17
F1_000145 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
ram_bank = 1
```

```
        clrf keytemp
F1_000146 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000148 equ $ ; IN [MITFPGUN.BAS] SELECT CASE KEYTEMP
F1_000150 equ $ ; IN [MITFPGUN.BAS] CASE "1"
        movlw 49
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL20
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL20
F1_000152 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000153 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"1.CHANGE ADMIN FP"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB10)
        movwf GEN4H
        movlw low (STR@LB10)
        f@call C@STROUT
F1_000154 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"2.CHANGE USER FP"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB11)
        movwf GEN4H
        movlw low (STR@LB11)
        f@call C@STROUT
F1_000155 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"3.CANCEL"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB12)
        movwf GEN4H
        movlw low (STR@LB12)
        f@call C@STROUT
```

```
F1_000157 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL22
F1_000159 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
ram_bank = 1
        clrf keytemp
F1_000161 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000163 equ $ ; IN [MITFPGUN.BAS] SELECT CASE KEYTEMP
F1_000165 equ $ ; IN [MITFPGUN.BAS] CASE "1","2"
        movlw 49
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL26
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,2
        goto BC@LL26
        movlw 50
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL25
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL25
BC@LL26
F1_000167 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000168 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"DELETING FP"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB13)
        movwf GEN4H
        movlw low (STR@LB13)
        f@call C@STROUT
F1_000169 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"PLEASE WAIT.."
        movlw 128
        movwf BPFH
```

```
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB14)
        movwf GEN4H
        movlw low (STR@LB14)
        f@call C@STROUT
F1_000172 equ $ ; IN [MITFPGUN.BAS] USERID = KEYTEMP - $30
        movlw 208
        bsf STATUS,5
ram_bank = 1
        addwf keytemp,W
        movwf userID
        movlw 255
        btfsc STATUS,0
        addlw 1
        movwf userIDH
F1_000173 equ $ ; IN [MITFPGUN.BAS] FPC = 2
        movlw 2
        bcf STATUS,5
ram_bank = 0
        movwf FPC
F1_000174 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCOMMAND
        f@call subs_FPCommand
F1_000176 equ $ ; IN [MITFPGUN.BAS] DELAYMS 1000
        movlw 3
        movwf PP1H
        movlw 232
        f@call DLY@W
F1_000178 equ $ ; IN [MITFPGUN.BAS] USERID = KEYTEMP - $30
        movlw 208
        bsf STATUS,5
ram_bank = 1
        addwf keytemp,W
        movwf userID
        movlw 255
        btfsc STATUS,0
        addlw 1
        movwf userIDH
F1_000179 equ $ ; IN [MITFPGUN.BAS] FPC = 1
        movlw 1
        bcf STATUS,5
ram_bank = 0
        movwf FPC
F1_000180 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCOMMAND
```

```
        f@call subs_FPCommand
F1_000182 equ $ ; IN [MITFPGUN.BAS] GOTO JMP_ADMINSTART
        F@JUMP jmp_AdminStart
        F@JUMP BC@LL24
BC@LL25
F1_000184 equ $ ; IN [MITFPGUN.BAS] CASE "3"
        movlw 51
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL28
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL28
F1_000186 equ $ ; IN [MITFPGUN.BAS] GOTO JMP_ADMINSTART
        F@JUMP jmp_AdminStart
F1_000188 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL28
BC@LL24
F1_000190 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL22
BC@LL23
        F@JUMP BC@LL19
BC@LL20
F1_000193 equ $ ; IN [MITFPGUN.BAS] CASE "2"
        movlw 50
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL30
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL30
F1_000195 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_SMS_REGISTERUSER
        f@call subs_SMS_RegisterUser
F1_000196 equ $ ; IN [MITFPGUN.BAS] GOTO JMP_ADMINSTART
        F@JUMP jmp_AdminStart
        F@JUMP BC@LL19
BC@LL30
F1_000198 equ $ ; IN [MITFPGUN.BAS] CASE "3"
        movlw 51
        bsf STATUS,5
```

```
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL32
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL32
F1_000200 equ $ ; IN [MITFPGUN.BAS] GOTO JMPFIRINGMODE
        F@JUMP jmpFiringMode
        F@JUMP BC@LL19
BC@LL32
F1_000202 equ $ ; IN [MITFPGUN.BAS] CASE "4"
        movlw 52
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL34
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL34
F1_000204 equ $ ; IN [MITFPGUN.BAS] GOTO MAINCODE
        F@JUMP mainCode
F1_000206 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL34
BC@LL19
F1_000208 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL17
BC@LL18
        F@JUMP BC@LL9
BC@LL16
F1_000210 equ $ ; IN [MITFPGUN.BAS] CASE 2
        bsf STATUS,5
ram_bank = 1
        movf userID,W
        xorlw 2
        iorwf userIDH,W
        set@page BC@LL36
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL36
F1_000214 equ $ ; IN [MITFPGUN.BAS] GOTO JMPFIRINGMODE
        F@JUMP jmpFiringMode
```

```
F1_000216 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL36
BC@LL9
F1_000219 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL8
F1_000221 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL5
BC@LL6
jmpFiringMode
F1_000225 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_UNLOCKGUN
        f@call subs_UnLockGun
F1_000226 equ $ ; IN [MITFPGUN.BAS] SHOTCTR = 0
        bsf STATUS,5
ram_bank = 1
        clrf shotCtr
F1_000228 equ $ ; IN [MITFPGUN.BAS] CLS
        bcf STATUS,5
ram_bank = 0
        f@call LCD@CLS
F1_000229 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"     SMART GUN"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB15)
        movwf GEN4H
        movlw low (STR@LB15)
        f@call C@STROUT
F1_000230 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"  PRESS # TO LOCK"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB16)
        movwf GEN4H
        movlw low (STR@LB16)
        f@call C@STROUT
F1_000231 equ $ ; IN [MITFPGUN.BAS] PRINT AT 4,1,"SHOTS: "
        movlw 128
        movwf BPFH
        movlw 212
        f@call LCD@CRS
        movlw high (STR@LB17)
        movwf GEN4H
        movlw low (STR@LB17)
```

```
        f@call C@STROUT
F1_000233 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL37
F1_000235 equ $ ; IN [MITFPGUN.BAS] PRINT AT 4,8, DEC3 SHOTCTR
        movlw 128
        movwf BPFH
        movlw 219
        f@call LCD@CRS
        movlw 3
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf shotCtr,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
F1_000237 equ $ ; IN [MITFPGUN.BAS] KPCOL1 = 1
        bsf PORTB,4
F1_000238 equ $ ; IN [MITFPGUN.BAS] KPCOL2 = 1
        bsf PORTB,5
F1_000239 equ $ ; IN [MITFPGUN.BAS] KPCOL3 = 0
        bcf PORTB,6
F1_000241 equ $ ; IN [MITFPGUN.BAS] IF KPROW4 = 0 THEN
        set@page BC@LL40
        btfsc PORTB,3
        goto BC@LL40
F1_000243 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000245 equ $ ; IN [MITFPGUN.BAS] WHILE KPROW1 = 0 : WEND
BC@LL41
        set@page BC@LL42
        btfsc PORTB,0
        goto BC@LL42
ram_bank = 0
        F@JUMP BC@LL41
BC@LL42
F1_000247 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000249 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1, "GUN IS LOCKED.      "
        movlw 128
        movwf BPFH
        movlw 148
```

```
        f@call LCD@CRS
        movlw high (STR@LB18)
        movwf GEN4H
        movlw low (STR@LB18)
        f@call C@STROUT
F1_000251 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_SENDDATA
        f@call subs_SendData
F1_000253 equ $ ; IN [MITFPGUN.BAS] GOTO MAINCODE
        F@JUMP mainCode
        F@JUMP BC@LL39
F1_000255 equ $ ; IN [MITFPGUN.BAS] ELSEIF TRIGTS = 0 THEN
BC@LL40
        set@page BC@LL43
        btfsc PORTD,6
        goto BC@LL43
F1_000257 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000259 equ $ ; IN [MITFPGUN.BAS] WHILE TRIGTS = 0 : WEND
BC@LL44
        set@page BC@LL45
        btfsc PORTD,6
        goto BC@LL45
ram_bank = 0
        F@JUMP BC@LL44
BC@LL45
F1_000261 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000263 equ $ ; IN [MITFPGUN.BAS] INC SHOTCTR
        bsf STATUS,5
ram_bank = 1
        incf shotCtr,F
F1_000266 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL43
        bcf STATUS,5
ram_bank = 0
BC@LL39
F1_000268 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL37
BC@LL38
F1_000276 equ $ ; IN [MITFPGUN.BAS] GOTO MAINCODE
        F@JUMP mainCode
subs_FPChkSum
```

```
F1_000281 equ $ ; IN [MITFPGUN.BAS] CHKTMP = 0
        bsf STATUS,5
ram_bank = 1
        clrf chkTmpH
        clrf chkTmp
F1_000283 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 0 TO 21
        clrf gCtr
FR@LB47
        bcf STATUS,5
ram_bank = 0
        movlw 22
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB48
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB48
F1_000285 equ $ ; IN [MITFPGUN.BAS] CHKTMP = CHKTMP + SDATA[GCTR]
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw sData
        movwf FSR
        movf INDF,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        addwf chkTmp,F
        btfsc STATUS,0
        incf chkTmpH,F
CT@LB49
        bcf STATUS,5
ram_bank = 0
F1_000287 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB47
```

```
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB47
NX@LB48
F1_000289 equ $ ; IN [MITFPGUN.BAS] SDATA[22] = CHKSLOW
        bsf STATUS,5
ram_bank = 1
        movf chkTmp,W
        bcf STATUS,5
ram_bank = 0
        movwf sData#22
F1_000290 equ $ ; IN [MITFPGUN.BAS] SDATA[23] = CHKSHI
        bsf STATUS,5
ram_bank = 1
        movf chkTmpH,W
        bcf STATUS,5
ram_bank = 0
        movwf sData#23
F1_000292 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_FPCommand
F1_000296 equ $ ; IN [MITFPGUN.BAS] SELECT CASE FPC
F1_000299 equ $ ; IN [MITFPGUN.BAS] CASE 0
        movf FPC,F
        set@page BC@LL50
        btfss STATUS,2
        goto BC@LL50
F1_000301 equ $ ; IN [MITFPGUN.BAS] SDATA[0] = $55
        movlw 85
        movwf sData#0
F1_000302 equ $ ; IN [MITFPGUN.BAS] SDATA[1] = $AA
        movlw 170
        movwf sData#1
F1_000303 equ $ ; IN [MITFPGUN.BAS] SDATA[2] = $02
        movlw 2
        movwf sData#2
F1_000304 equ $ ; IN [MITFPGUN.BAS] SDATA[3] = $01
        movlw 1
        movwf sData#3
F1_000306 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 4 TO 21
        movlw 4
        bsf STATUS,5
ram_bank = 1
```

```
        movwf gCtr
FR@LB53
        bcf STATUS,5
ram_bank = 0
        movlw 22
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB54
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB54
F1_000308 equ $ ; IN [MITFPGUN.BAS] SDATA[GCTR] = $00
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw sData
        movwf FSR
        movlw 0
        movwf INDF
CT@LB55
        bcf STATUS,5
ram_bank = 0
F1_000310 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB53
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB53
NX@LB54
F1_000312 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCHKSUM
        f@call subs_FPChkSum
F1_000314 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000315 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"VERIFYING"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB19)
```

```
        movwf GEN4H
        movlw low (STR@LB19)
        f@call C@STROUT
F1_000316 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"FINGERPRINT"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB20)
        movwf GEN4H
        movlw low (STR@LB20)
        f@call C@STROUT
F1_000318 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPSENDRQ
        f@call subs_FPSendRQ
F1_000320 equ $ ; IN [MITFPGUN.BAS] USERID = 0
        bsf STATUS,5
ram_bank = 1
        clrf userIDH
        clrf userID
F1_000322 equ $ ; IN [MITFPGUN.BAS] IF SREPLY[8] = $FF AND SREPLY[7] = $F4 THEN
        bcf STATUS,5
ram_bank = 0
        movf sReply#8,W
        movwf PBP#VAR0
        incf PBP#VAR0,W
        movlw 1
        btfss STATUS,2
        movlw 0
        movwf SP#P9
        movf sReply#7,W
        movwf PBP#VAR0
        movlw 244
        subwf PBP#VAR0,W
        movlw 1
        btfss STATUS,2
        movlw 0
        andwf SP#P9,F
        set@page BC@LL56
        btfsc STATUS,2
        goto BC@LL56
F1_000324 equ $ ; IN [MITFPGUN.BAS] SERIN PORTD.5, 84,[WAIT($AA),STR SREPLY\23]
        movlw PORTD
        movwf GEN
        movlw 32
```

```
        movwf GENH
        clrf PP5H
        movlw 84
        movwf PP5
PB@LB58
        f@call SERIN
        sublw 170
        set@page PB@LB58
        btfss STATUS,2
        goto PB@LB58
        clrf PBP#VAR0
PB@LB61
        f@call SERIN
        movf PBP#VAR0,W
        movwf GEN4H
        movlw 119
        movwf GEN4
        movf PP1,W
        f@call PUT@ARRAY
        incf PBP#VAR0,F
        movf PBP#VAR0,W
        sublw 23
        set@page PB@LB61
        btfss STATUS,2
        goto PB@LB61
F1_000326 equ $ ; IN [MITFPGUN.BAS] IF SREPLY[5] = $00 THEN
        movf sReply#5,W
        movwf PBP#VAR0
        movf PBP#VAR0,F
        set@page BC@LL62
        btfss STATUS,2
        goto BC@LL62
F1_000328 equ $ ; IN [MITFPGUN.BAS] UIDL = SREPLY[7]
        movf sReply#7,W
        bsf STATUS,5
ram_bank = 1
        movwf userID
F1_000329 equ $ ; IN [MITFPGUN.BAS] UIDH = SREPLY[8]
        bcf STATUS,5
ram_bank = 0
        movf sReply#8,W
        bsf STATUS,5
ram_bank = 1
        movwf userIDH
```

```
F1_000331 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL62
        bcf STATUS,5
ram_bank = 0
F1_000333 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL56
        F@JUMP BC@LL49
BC@LL50
F1_000335 equ $ ; IN [MITFPGUN.BAS] CASE 1
        movlw 1
        subwf FPC,W
        set@page BC@LL64
        btfss STATUS,2
        goto BC@LL64
F1_000337 equ $ ; IN [MITFPGUN.BAS] SDATA[0] = $55
        movlw 85
        movwf sData#0
F1_000338 equ $ ; IN [MITFPGUN.BAS] SDATA[1] = $AA
        movlw 170
        movwf sData#1
F1_000339 equ $ ; IN [MITFPGUN.BAS] SDATA[2] = $03
        movlw 3
        movwf sData#2
F1_000340 equ $ ; IN [MITFPGUN.BAS] SDATA[3] = $01
        movlw 1
        movwf sData#3
F1_000341 equ $ ; IN [MITFPGUN.BAS] SDATA[4] = $02
        movlw 2
        movwf sData#4
F1_000342 equ $ ; IN [MITFPGUN.BAS] SDATA[5] = $00
        clrf sData#5
F1_000343 equ $ ; IN [MITFPGUN.BAS] SDATA[6] = UIDL
        bsf STATUS,5
ram_bank = 1
        movf userID,W
        bcf STATUS,5
ram_bank = 0
        movwf sData#6
F1_000344 equ $ ; IN [MITFPGUN.BAS] SDATA[7] = UIDH
        bsf STATUS,5
ram_bank = 1
        movf userIDH,W
        bcf STATUS,5
ram_bank = 0
```

```
        movwf sData#7
F1_000346 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 8 TO 21
        movlw 8
        bsf STATUS,5
ram_bank = 1
        movwf gCtr
FR@LB66
        bcf STATUS,5
ram_bank = 0
        movlw 22
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB67
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB67
F1_000348 equ $ ; IN [MITFPGUN.BAS] SDATA[GCTR] = $00
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw sData
        movwf FSR
        movlw 0
        movwf INDF
CT@LB68
        bcf STATUS,5
ram_bank = 0
F1_000350 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB66
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB66
NX@LB67
F1_000352 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCHKSUM
        f@call subs_FPChkSum
F1_000354 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
```

```
F1_000355 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"ENROLLING FP"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB21)
        movwf GEN4H
        movlw low (STR@LB21)
        f@call C@STROUT
F1_000356 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"TAKING SAMPLE# 1"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB22)
        movwf GEN4H
        movlw low (STR@LB22)
        f@call C@STROUT
F1_000358 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPSENDRQ
        f@call subs_FPSendRQ
F1_000360 equ $ ; IN [MITFPGUN.BAS] GCTR = 1
        movlw 1
        bsf STATUS,5
ram_bank = 1
        movwf gCtr
F1_000363 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL68
        bcf STATUS,5
ram_bank = 0
F1_000366 equ $ ; IN [MITFPGUN.BAS] IF SREPLY[5] = $00 THEN
        movf sReply#5,W
        movwf PBP#VAR0
        movf PBP#VAR0,F
        set@page BC@LL71
        btfss STATUS,2
        goto BC@LL71
F1_000368 equ $ ; IN [MITFPGUN.BAS] IF SREPLY[8] = $FF THEN
        movf sReply#8,W
        movwf PBP#VAR0
        incf PBP#VAR0,W
        set@page BC@LL73
        btfss STATUS,2
        goto BC@LL73
F1_000370 equ $ ; IN [MITFPGUN.BAS] SELECT CASE SREPLY[7]
        movf sReply#7,W
```

```
        movwf PBP#VAR0
F1_000372 equ $ ; IN [MITFPGUN.BAS] CASE $F1, $F2, $F3
        movlw 241
        subwf PBP#VAR0,W
        set@page BC@LL76
        btfsc STATUS,2
        goto BC@LL76
        movlw 242
        subwf PBP#VAR0,W
        set@page BC@LL76
        btfsc STATUS,2
        goto BC@LL76
        movlw 243
        subwf PBP#VAR0,W
        set@page BC@LL75
        btfss STATUS,2
        goto BC@LL75
BC@LL76
F1_000374 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"TAKING SAMPLE# ", DEC GCTR,"
"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB23)
        movwf GEN4H
        movlw low (STR@LB23)
        f@call C@STROUT
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECB
        movlw 32
        f@call PRINT
        f@call PRINT
        f@call PRINT
        f@call PRINT
F1_000376 equ $ ; IN [MITFPGUN.BAS] IF GCTR < 4 THEN
        movlw 4
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
```

```
        set@page BC@LL78
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto BC@LL78
F1_000378 equ $ ; IN [MITFPGUN.BAS] INC GCTR
        bsf STATUS,5
ram_bank = 1
        incf gCtr,F
        bcf STATUS,5
ram_bank = 0
        F@JUMP BC@LL79
BC@LL78
F1_000380 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000382 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_FPC_COMPLETE
        F@JUMP sj_FPC_Complete
F1_000384 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL79
        F@JUMP BC@LL74
BC@LL75
F1_000386 equ $ ; IN [MITFPGUN.BAS] CASE $F4
        movlw 244
        subwf PBP#VAR0,W
        set@page BC@LL81
        btfss STATUS,2
        goto BC@LL81
F1_000388 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"LIFT FINGER        "
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB24)
        movwf GEN4H
        movlw low (STR@LB24)
        f@call C@STROUT
F1_000391 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL81
BC@LL74
        F@JUMP BC@LL72
F1_000393 equ $ ; IN [MITFPGUN.BAS] ELSEIF SREPLY[7] = $21 AND SREPLY[8] = $00
THEN
BC@LL73
        movf sReply#7,W
        movwf PBP#VAR1
```

```
        movlw 33
        subwf PBP#VAR1,W
        movlw 1
        btfss STATUS,2
        movlw 0
        movwf SP#P9
        movf sReply#8,W
        movwf PBP#VAR1
        movlw 1
        movf PBP#VAR1,F
        btfss STATUS,2
        movlw 0
        andwf SP#P9,F
        set@page BC@LL82
        btfsc STATUS,2
        goto BC@LL82
F1_000395 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"SAMPLING ERROR      "
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB25)
        movwf GEN4H
        movlw low (STR@LB25)
        f@call C@STROUT
        F@JUMP BC@LL72
F1_000397 equ $ ; IN [MITFPGUN.BAS] ELSEIF SREPLY[3] = $06 THEN
BC@LL82
        movf sReply#3,W
        movwf PBP#VAR1
        movlw 6
        subwf PBP#VAR1,W
        set@page BC@LL83
        btfss STATUS,2
        goto BC@LL83
F1_000399 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"SAMPLING COMPLETE   "
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB26)
        movwf GEN4H
        movlw low (STR@LB26)
        f@call C@STROUT
```

```
F1_000400 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_FPC_COMPLETE
        F@JUMP sj_FPC_Complete
F1_000402 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL83
BC@LL72
        F@JUMP BC@LL84
BC@LL71
F1_000404 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000406 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"SAMPLING ERROR      "
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB27)
        movwf GEN4H
        movlw low (STR@LB27)
        f@call C@STROUT
F1_000407 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_FPC_COMPLETE
        F@JUMP sj_FPC_Complete
F1_000409 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL84
F1_000411 equ $ ; IN [MITFPGUN.BAS] SERIN PORTD.5, 84,[WAIT($AA),STR SREPLY\23]
        movlw PORTD
        movwf GEN
        movlw 32
        movwf GENH
        clrf PP5H
        movlw 84
        movwf PP5
PB@LB86
        f@call SERIN
        sublw 170
        set@page PB@LB86
        btfss STATUS,2
        goto PB@LB86
        clrf PBP#VAR2
PB@LB89
        f@call SERIN
        movf PBP#VAR2,W
        movwf GEN4H
        movlw 119
        movwf GEN4
        movf PP1,W
        f@call PUT@ARRAY
```

```
        incf PBP#VAR2,F
        movf PBP#VAR2,W
        sublw 23
        set@page PB@LB89
        btfss STATUS,2
        goto PB@LB89
F1_000413 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL68
BC@LL69
        F@JUMP BC@LL49
BC@LL64
F1_000417 equ $ ; IN [MITFPGUN.BAS] CASE 2
        movlw 2
        subwf FPC,W
        set@page BC@LL90
        btfss STATUS,2
        goto BC@LL90
F1_000419 equ $ ; IN [MITFPGUN.BAS] SDATA[0] = $55
        movlw 85
        movwf sData#0
F1_000420 equ $ ; IN [MITFPGUN.BAS] SDATA[1] = $AA
        movlw 170
        movwf sData#1
F1_000421 equ $ ; IN [MITFPGUN.BAS] SDATA[2] = $05
        movlw 5
        movwf sData#2
F1_000422 equ $ ; IN [MITFPGUN.BAS] SDATA[3] = $01
        movlw 1
        movwf sData#3
F1_000423 equ $ ; IN [MITFPGUN.BAS] SDATA[4] = $02
        movlw 2
        movwf sData#4
F1_000424 equ $ ; IN [MITFPGUN.BAS] SDATA[5] = $00
        clrf sData#5
F1_000425 equ $ ; IN [MITFPGUN.BAS] SDATA[6] = UIDL
        bsf STATUS,5
ram_bank = 1
        movf userID,W
        bcf STATUS,5
ram_bank = 0
        movwf sData#6
F1_000426 equ $ ; IN [MITFPGUN.BAS] SDATA[7] = UIDH
        bsf STATUS,5
ram_bank = 1
```

```
        movf userIDH,W
        bcf STATUS,5
ram_bank = 0
        movwf sData#7
F1_000428 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 8 TO 21
        movlw 8
        bsf STATUS,5
ram_bank = 1
        movwf gCtr
FR@LB92
        bcf STATUS,5
ram_bank = 0
        movlw 22
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB93
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB93
F1_000430 equ $ ; IN [MITFPGUN.BAS] SDATA[GCTR] = $00
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw sData
        movwf FSR
        movlw 0
        movwf INDF
CT@LB94
        bcf STATUS,5
ram_bank = 0
F1_000432 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB92
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB92
NX@LB93
F1_000434 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCHKSUM
```

```
        f@call subs_FPChkSum
F1_000436 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000437 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"DELETING.."
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB28)
        movwf GEN4H
        movlw low (STR@LB28)
        f@call C@STROUT
F1_000439 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPSENDRQ
        f@call subs_FPSendRQ
F1_000441 equ $ ; IN [MITFPGUN.BAS] IF SREPLY[5] = $00 THEN
        movf sReply#5,W
        movwf PBP#VAR2
        movf PBP#VAR2,F
        set@page BC@LL95
        btfss STATUS,2
        goto BC@LL95
F1_000443 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"USER DELETED.      "
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB29)
        movwf GEN4H
        movlw low (STR@LB29)
        f@call C@STROUT
        F@JUMP BC@LL96
BC@LL95
F1_000445 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000447 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"OPERATION ERROR    "
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB30)
        movwf GEN4H
        movlw low (STR@LB30)
        f@call C@STROUT
F1_000448 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_FPC_COMPLETE
        F@JUMP sj_FPC_Complete
F1_000450 equ $ ; IN [MITFPGUN.BAS] ENDIF
```

```
BC@LL96
F1_000453 equ $ ; IN [MITFPGUN.BAS] END SELECT
BC@LL90
BC@LL49
sj_FPC_Complete
F1_000461 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_FPSendRQ
sj_FPSRQ
F1_000467 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 0 TO 23
        bsf STATUS,5
ram_bank = 1
        clrf gCtr
FR@LB98
        bcf STATUS,5
ram_bank = 0
        movlw 24
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB99
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB99
F1_000469 equ $ ; IN [MITFPGUN.BAS] SREPLY[GCTR] = $00
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        bcf STATUS,5
ram_bank = 0
        movwf GEN4H
        movlw 119
        movwf GEN4
        movlw 0
        f@call PUT@ARRAY
CT@LB100
F1_000471 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB98
        bcf STATUS,5
```

```
ram_bank = 0
        btfss STATUS,0
        goto FR@LB98
NX@LB99
F1_000473 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 0 TO 23
        bsf STATUS,5
ram_bank = 1
        clrf gCtr
FR@LB101
        bcf STATUS,5
ram_bank = 0
        movlw 24
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB102
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB102
F1_000475 equ $ ; IN [MITFPGUN.BAS] SEROUT PORTD.4, 84,[SDATA[GCTR]]
        movlw PORTD
        movwf GEN
        movlw 16
        movwf GENH
        clrf PP5H
        movlw 84
        movwf PP5
        clrf GEN3H
        clrf GEN3
        movlw 32
        movwf BPFH
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw sData
        movwf FSR
        movf INDF,W
        bcf STATUS,5
ram_bank = 0
        f@call SEROUT
CT@LB103
F1_000477 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
```

```
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB101
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB101
NX@LB102
F1_000479 equ $ ; IN [MITFPGUN.BAS] SERIN PORTD.5, 84,[WAIT($AA),STR SREPLY\23]
        movlw PORTD
        movwf GEN
        movlw 32
        movwf GENH
        clrf PP5H
        movlw 84
        movwf PP5
PB@LB104
        f@call SERIN
        sublw 170
        set@page PB@LB104
        btfss STATUS,2
        goto PB@LB104
        clrf PBP#VAR0
PB@LB107
        f@call SERIN
        movf PBP#VAR0,W
        movwf GEN4H
        movlw 119
        movwf GEN4
        movf PP1,W
        f@call PUT@ARRAY
        incf PBP#VAR0,F
        movf PBP#VAR0,W
        sublw 23
        set@page PB@LB107
        btfss STATUS,2
        goto PB@LB107
F1_000482 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_KP3x416FR
F1_000486 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL107
F1_000488 equ $ ; IN [MITFPGUN.BAS] FOR KPCTR = 0 TO 2
```

```
        bsf STATUS,5
ram_bank = 1
        clrf kpCtr
FR@LB110
        bcf STATUS,5
ram_bank = 0
        movlw 3
        bsf STATUS,5
ram_bank = 1
        subwf kpCtr,W
        set@page NX@LB111
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB111
F1_000490 equ $ ; IN [MITFPGUN.BAS] SELECT CASE KPCTR
F1_000492 equ $ ; IN [MITFPGUN.BAS] CASE 0
        bsf STATUS,5
ram_bank = 1
        movf kpCtr,F
        set@page BC@LL113
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL113
F1_000494 equ $ ; IN [MITFPGUN.BAS] KPCOL1 = 0
        bcf PORTB,4
F1_000495 equ $ ; IN [MITFPGUN.BAS] KPCOL2 = 1
        bsf PORTB,5
F1_000496 equ $ ; IN [MITFPGUN.BAS] KPCOL3 = 1
        bsf PORTB,6
        F@JUMP BC@LL112
BC@LL113
F1_000498 equ $ ; IN [MITFPGUN.BAS] CASE 1
        movlw 1
        bsf STATUS,5
ram_bank = 1
        subwf kpCtr,W
        set@page BC@LL116
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL116
F1_000500 equ $ ; IN [MITFPGUN.BAS] KPCOL1 = 1
```

```
        bsf PORTB,4
F1_000501 equ $ ; IN [MITFPGUN.BAS] KPCOL2 = 0
        bcf PORTB,5
F1_000502 equ $ ; IN [MITFPGUN.BAS] KPCOL3 = 1
        bsf PORTB,6
        F@JUMP BC@LL112
BC@LL116
F1_000504 equ $ ; IN [MITFPGUN.BAS] CASE 2
        movlw 2
        bsf STATUS,5
ram_bank = 1
        subwf kpCtr,W
        set@page BC@LL118
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL118
F1_000506 equ $ ; IN [MITFPGUN.BAS] KPCOL1 = 1
        bsf PORTB,4
F1_000507 equ $ ; IN [MITFPGUN.BAS] KPCOL2 = 1
        bsf PORTB,5
F1_000508 equ $ ; IN [MITFPGUN.BAS] KPCOL3 = 0
        bcf PORTB,6
F1_000510 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL118
BC@LL112
F1_000512 equ $ ; IN [MITFPGUN.BAS] IF KPROW1 = 0 THEN
        set@page BC@LL120
        btfsc PORTB,0
        goto BC@LL120
F1_000514 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000516 equ $ ; IN [MITFPGUN.BAS] WHILE KPROW1 = 0 : WEND
BC@LL121
        set@page BC@LL122
        btfsc PORTB,0
        goto BC@LL122
ram_bank = 0
        F@JUMP BC@LL121
BC@LL122
F1_000518 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
```

```
F1_000520 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = LOOKUPL KPCTR,["1","2","3"]
        bsf STATUS,5
ram_bank = 1
        movf kpCtr,W
        bcf STATUS,5
ram_bank = 0
        movwf PP1
        sublw 2
        set@page LE@LB1
        btfss STATUS,0
        goto LE@LB1
        bcf STATUS,0
        rlf PP1,W
        addwf PP1,F
        f@call L@LB3
        movwf PP0
        incf PP1,F
        f@call L@LB3
        movwf PP0H
        incf PP1,F
        f@call L@LB3
        f@call L@KL
        movf PP0,W
        bsf STATUS,5
ram_bank = 1
        movwf keytemp
        bcf STATUS,5
ram_bank = 0
        F@JUMP LE@LB1
L@LB3
        movlw ((LK@LB2 >> 8) & 255)
        movwf PCLATH
        movf PP1,W
        if($ >> 8) !=(($ + 9) >> 8)
        goto SO@LB123
        org(($ + 255) & 65280)
SO@LB123
        endif
        addwf PCL,F
LK@LB2
        dt (49 & 255),((49 >> 8) & 255),0
        dt (50 & 255),((50 >> 8) & 255),0
        dt (51 & 255),((51 >> 8) & 255),0
LE@LB1
```

```
F1_000522 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_KPR_END
        F@JUMP sj_kpr_end
        F@JUMP BC@LL119
F1_000524 equ $ ; IN [MITFPGUN.BAS] ELSEIF KPROW2 = 0 THEN
BC@LL120
        set@page BC@LL124
        btfsc PORTB,1
        goto BC@LL124
F1_000526 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000528 equ $ ; IN [MITFPGUN.BAS] WHILE KPROW2 = 0 : WEND
BC@LL125
        set@page BC@LL126
        btfsc PORTB,1
        goto BC@LL126
ram_bank = 0
        F@JUMP BC@LL125
BC@LL126
F1_000530 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000532 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = LOOKUPL KPCTR,["4","5","6"]
        bsf STATUS,5
ram_bank = 1
        movf kpCtr,W
        bcf STATUS,5
ram_bank = 0
        movwf PP1
        sublw 2
        set@page LE@LB4
        btfss STATUS,0
        goto LE@LB4
        bcf STATUS,0
        rlf PP1,W
        addwf PP1,F
        f@call L@LB6
        movwf PP0
        incf PP1,F
        f@call L@LB6
        movwf PP0H
        incf PP1,F
        f@call L@LB6
        f@call L@KL
```

```
        movf PP0,W
        bsf STATUS,5
ram_bank = 1
        movwf keytemp
        bcf STATUS,5
ram_bank = 0
        F@JUMP LE@LB4
L@LB6
        movlw ((LK@LB5 >> 8) & 255)
        movwf PCLATH
        movf PP1,W
        if($ >> 8) !=(($ + 9) >> 8)
        goto SO@LB127
        org(($ + 255) & 65280)
SO@LB127
        endif
        addwf PCL,F
LK@LB5
        dt (52 & 255),((52 >> 8) & 255),0
        dt (53 & 255),((53 >> 8) & 255),0
        dt (54 & 255),((54 >> 8) & 255),0
LE@LB4
F1_000534 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_KPR_END
        F@JUMP sj_kpr_end
        F@JUMP BC@LL119
F1_000536 equ $ ; IN [MITFPGUN.BAS] ELSEIF KPROW3 = 0 THEN
BC@LL124
        set@page BC@LL128
        btfsc PORTB,2
        goto BC@LL128
F1_000538 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000540 equ $ ; IN [MITFPGUN.BAS] WHILE KPROW3 = 0 : WEND
BC@LL129
        set@page BC@LL130
        btfsc PORTB,2
        goto BC@LL130
ram_bank = 0
        F@JUMP BC@LL129
BC@LL130
F1_000542 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
```

```
F1_000544 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = LOOKUPL KPCTR,["7","8","9"]
        bsf STATUS,5
ram_bank = 1
        movf kpCtr,W
        bcf STATUS,5
ram_bank = 0
        movwf PP1
        sublw 2
        set@page LE@LB7
        btfss STATUS,0
        goto LE@LB7
        bcf STATUS,0
        rlf PP1,W
        addwf PP1,F
        f@call L@LB9
        movwf PP0
        incf PP1,F
        f@call L@LB9
        movwf PP0H
        incf PP1,F
        f@call L@LB9
        f@call L@KL
        movf PP0,W
        bsf STATUS,5
ram_bank = 1
        movwf keytemp
        bcf STATUS,5
ram_bank = 0
        F@JUMP LE@LB7
L@LB9
        movlw ((LK@LB8 >> 8) & 255)
        movwf PCLATH
        movf PP1,W
        if($ >> 8) !=(($ + 9) >> 8)
        goto SO@LB131
        org(($ + 255) & 65280)
SO@LB131
        endif
        addwf PCL,F
LK@LB8
        dt (55 & 255),((55 >> 8) & 255),0
        dt (56 & 255),((56 >> 8) & 255),0
        dt (57 & 255),((57 >> 8) & 255),0
LE@LB7
```

```
F1_000546 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_KPR_END
        F@JUMP sj_kpr_end
        F@JUMP BC@LL119
F1_000548 equ $ ; IN [MITFPGUN.BAS] ELSEIF KPROW4 = 0 THEN
BC@LL128
        set@page BC@LL132
        btfsc PORTB,3
        goto BC@LL132
F1_000550 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000552 equ $ ; IN [MITFPGUN.BAS] WHILE KPROW4 = 0: WEND
BC@LL133
        set@page BC@LL134
        btfsc PORTB,3
        goto BC@LL134
ram_bank = 0
        F@JUMP BC@LL133
BC@LL134
F1_000554 equ $ ; IN [MITFPGUN.BAS] DELAYMS 50
        movlw 50
        f@call DL@MS
F1_000556 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = LOOKUPL KPCTR,["*","0","#"]
        bsf STATUS,5
ram_bank = 1
        movf kpCtr,W
        bcf STATUS,5
ram_bank = 0
        movwf PP1
        sublw 2
        set@page LE@LB10
        btfss STATUS,0
        goto LE@LB10
        bcf STATUS,0
        rlf PP1,W
        addwf PP1,F
        f@call L@LB12
        movwf PP0
        incf PP1,F
        f@call L@LB12
        movwf PP0H
        incf PP1,F
        f@call L@LB12
        f@call L@KL
```

```
        movf PP0,W
        bsf STATUS,5
ram_bank = 1
        movwf keytemp
        bcf STATUS,5
ram_bank = 0
        F@JUMP LE@LB10
L@LB12
        movlw ((LK@LB11 >> 8) & 255)
        movwf PCLATH
        movf PP1,W
        if($ >> 8) !=(($ + 9) >> 8)
        goto SO@LB135
        org(($ + 255) & 65280)
SO@LB135
        endif
        addwf PCL,F
LK@LB11
        dt (42 & 255),((42 >> 8) & 255),0
        dt (48 & 255),((48 >> 8) & 255),0
        dt (35 & 255),((35 >> 8) & 255),0
LE@LB10
F1_000558 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_KPR_END
        F@JUMP sj_kpr_end
F1_000560 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL132
BC@LL119
CT@LB112
F1_000563 equ $ ; IN [MITFPGUN.BAS] NEXT KPCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf kpCtr,F
        set@page FR@LB110
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB110
NX@LB111
F1_000566 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL107
BC@LL108
sj_kpr_end
F1_000571 equ $ ; IN [MITFPGUN.BAS] RETURN
```

```
        return
subs_FPError
F1_000575 equ $ ; IN [MITFPGUN.BAS] SELECT CASE FPECODE
F1_000577 equ $ ; IN [MITFPGUN.BAS] CASE $00 : STR FPEMSG = "COMMAND
SUCCESFUL.  ",0
        movf FPECode,F
        set@page BC@LL137
        btfss STATUS,2
        goto BC@LL137
ram_bank = 0
        movlw 67
        movwf FPEMsg#0
        movlw 79
        movwf FPEMsg#1
        movlw 77
        movwf FPEMsg#2
        movwf FPEMsg#3
        movlw 65
        movwf FPEMsg#4
        movlw 78
        movwf FPEMsg#5
        movlw 68
        movwf FPEMsg#6
        movlw 32
        movwf FPEMsg#7
        movlw 83
        movwf FPEMsg#8
        movlw 85
        movwf FPEMsg#9
        movlw 67
        movwf FPEMsg#10
        movwf FPEMsg#11
        movlw 69
        movwf FPEMsg#12
        movlw 83
        movwf FPEMsg#13
        movlw 70
        movwf FPEMsg#14
        movlw 85
        movwf FPEMsg#15
        movlw 76
        movwf FPEMsg#16
        movlw 46
        movwf FPEMsg#17
```

```
        movlw 32
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL137
F1_000578 equ $ ; IN [MITFPGUN.BAS] CASE $01 : STR FPEMSG = "COMMAND FAILED.
",0
        movlw 1
        subwf FPECode,W
        set@page BC@LL140
        btfss STATUS,2
        goto BC@LL140
ram_bank = 0
        movlw 67
        movwf FPEMsg#0
        movlw 79
        movwf FPEMsg#1
        movlw 77
        movwf FPEMsg#2
        movwf FPEMsg#3
        movlw 65
        movwf FPEMsg#4
        movlw 78
        movwf FPEMsg#5
        movlw 68
        movwf FPEMsg#6
        movlw 32
        movwf FPEMsg#7
        movlw 70
        movwf FPEMsg#8
        movlw 65
        movwf FPEMsg#9
        movlw 73
        movwf FPEMsg#10
        movlw 76
        movwf FPEMsg#11
        movlw 69
        movwf FPEMsg#12
        movlw 68
        movwf FPEMsg#13
        movlw 46
        movwf FPEMsg#14
        movlw 32
```

```
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL140
F1_000579 equ $ ; IN [MITFPGUN.BAS] CASE $11 : STR FPEMSG = "MATCH NOT FOUND.
",0
        movlw 17
        subwf FPECode,W
        set@page BC@LL142
        btfss STATUS,2
        goto BC@LL142
ram_bank = 0
        movlw 77
        movwf FPEMsg#0
        movlw 65
        movwf FPEMsg#1
        movlw 84
        movwf FPEMsg#2
        movlw 67
        movwf FPEMsg#3
        movlw 72
        movwf FPEMsg#4
        movlw 32
        movwf FPEMsg#5
        movlw 78
        movwf FPEMsg#6
        movlw 79
        movwf FPEMsg#7
        movlw 84
        movwf FPEMsg#8
        movlw 32
        movwf FPEMsg#9
        movlw 70
        movwf FPEMsg#10
        movlw 79
        movwf FPEMsg#11
        movlw 85
        movwf FPEMsg#12
        movlw 78
        movwf FPEMsg#13
```

```
        movlw 68
        movwf FPEMsg#14
        movlw 46
        movwf FPEMsg#15
        movlw 32
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL142
F1_000580 equ $ ; IN [MITFPGUN.BAS] CASE $12 : STR FPEMSG = "ID NOT FOUND.
",0
        movlw 18
        subwf FPECode,W
        set@page BC@LL144
        btfss STATUS,2
        goto BC@LL144
ram_bank = 0
        movlw 73
        movwf FPEMsg#0
        movlw 68
        movwf FPEMsg#1
        movlw 32
        movwf FPEMsg#2
        movlw 78
        movwf FPEMsg#3
        movlw 79
        movwf FPEMsg#4
        movlw 84
        movwf FPEMsg#5
        movlw 32
        movwf FPEMsg#6
        movlw 70
        movwf FPEMsg#7
        movlw 79
        movwf FPEMsg#8
        movlw 85
        movwf FPEMsg#9
        movlw 78
        movwf FPEMsg#10
        movlw 68
        movwf FPEMsg#11
```

```
        movlw 46
        movwf FPEMsg#12
        movlw 32
        movwf FPEMsg#13
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL144
F1_000581 equ $ ; IN [MITFPGUN.BAS] CASE $13 : STR FPEMSG = "ID IS NULL  .      ",0
        movlw 19
        subwf FPECode,W
        set@page BC@LL146
        btfss STATUS,2
        goto BC@LL146
ram_bank = 0
        movlw 73
        movwf FPEMsg#0
        movlw 68
        movwf FPEMsg#1
        movlw 32
        movwf FPEMsg#2
        movlw 73
        movwf FPEMsg#3
        movlw 83
        movwf FPEMsg#4
        movlw 32
        movwf FPEMsg#5
        movlw 78
        movwf FPEMsg#6
        movlw 85
        movwf FPEMsg#7
        movlw 76
        movwf FPEMsg#8
        movwf FPEMsg#9
        movlw 32
        movwf FPEMsg#10
        movwf FPEMsg#11
        movlw 46
        movwf FPEMsg#12
```

```
        movlw 32
        movwf FPEMsg#13
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL146
F1_000582 equ $ ; IN [MITFPGUN.BAS] CASE $14 : STR FPEMSG = "ID ALREADY USED.
",0
        movlw 20
        subwf FPECode,W
        set@page BC@LL148
        btfss STATUS,2
        goto BC@LL148
ram_bank = 0
        movlw 73
        movwf FPEMsg#0
        movlw 68
        movwf FPEMsg#1
        movlw 32
        movwf FPEMsg#2
        movlw 65
        movwf FPEMsg#3
        movlw 76
        movwf FPEMsg#4
        movlw 82
        movwf FPEMsg#5
        movlw 69
        movwf FPEMsg#6
        movlw 65
        movwf FPEMsg#7
        movlw 68
        movwf FPEMsg#8
        movlw 89
        movwf FPEMsg#9
        movlw 32
        movwf FPEMsg#10
        movlw 85
        movwf FPEMsg#11
        movlw 83
```

```
        movwf FPEMsg#12
        movlw 69
        movwf FPEMsg#13
        movlw 68
        movwf FPEMsg#14
        movlw 46
        movwf FPEMsg#15
        movlw 32
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL148
F1_000583 equ $ ; IN [MITFPGUN.BAS] CASE $15 : STR FPEMSG = "NO REGISTERED
USERS.",0
        movlw 21
        subwf FPECode,W
        set@page BC@LL150
        btfss STATUS,2
        goto BC@LL150
ram_bank = 0
        movlw 78
        movwf FPEMsg#0
        movlw 79
        movwf FPEMsg#1
        movlw 32
        movwf FPEMsg#2
        movlw 82
        movwf FPEMsg#3
        movlw 69
        movwf FPEMsg#4
        movlw 71
        movwf FPEMsg#5
        movlw 73
        movwf FPEMsg#6
        movlw 83
        movwf FPEMsg#7
        movlw 84
        movwf FPEMsg#8
        movlw 69
        movwf FPEMsg#9
        movlw 82
```

```
        movwf FPEMsg#10
        movlw 69
        movwf FPEMsg#11
        movlw 68
        movwf FPEMsg#12
        movlw 32
        movwf FPEMsg#13
        movlw 85
        movwf FPEMsg#14
        movlw 83
        movwf FPEMsg#15
        movlw 69
        movwf FPEMsg#16
        movlw 82
        movwf FPEMsg#17
        movlw 83
        movwf FPEMsg#18
        movlw 46
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL150
F1_000584 equ $ ; IN [MITFPGUN.BAS] CASE $19 : STR FPEMSG = "ALREADY
REGISTERED. ",0
        movlw 25
        subwf FPECode,W
        set@page BC@LL152
        btfss STATUS,2
        goto BC@LL152
ram_bank = 0
        movlw 65
        movwf FPEMsg#0
        movlw 76
        movwf FPEMsg#1
        movlw 82
        movwf FPEMsg#2
        movlw 69
        movwf FPEMsg#3
        movlw 65
        movwf FPEMsg#4
        movlw 68
        movwf FPEMsg#5
        movlw 89
        movwf FPEMsg#6
```

```
        movlw 32
        movwf FPEMsg#7
        movlw 82
        movwf FPEMsg#8
        movlw 69
        movwf FPEMsg#9
        movlw 71
        movwf FPEMsg#10
        movlw 73
        movwf FPEMsg#11
        movlw 83
        movwf FPEMsg#12
        movlw 84
        movwf FPEMsg#13
        movlw 69
        movwf FPEMsg#14
        movlw 82
        movwf FPEMsg#15
        movlw 69
        movwf FPEMsg#16
        movlw 68
        movwf FPEMsg#17
        movlw 46
        movwf FPEMsg#18
        movlw 32
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL152
F1_000585 equ $ ; IN [MITFPGUN.BAS] CASE $21 : STR FPEMSG = "BAD IMAGE.          ",0
        movlw 33
        subwf FPECode,W
        set@page BC@LL154
        btfss STATUS,2
        goto BC@LL154
ram_bank = 0
        movlw 66
        movwf FPEMsg#0
        movlw 65
        movwf FPEMsg#1
        movlw 68
        movwf FPEMsg#2
        movlw 32
        movwf FPEMsg#3
```

```
        movlw 73
        movwf FPEMsg#4
        movlw 77
        movwf FPEMsg#5
        movlw 65
        movwf FPEMsg#6
        movlw 71
        movwf FPEMsg#7
        movlw 69
        movwf FPEMsg#8
        movlw 46
        movwf FPEMsg#9
        movlw 32
        movwf FPEMsg#10
        movwf FPEMsg#11
        movwf FPEMsg#12
        movwf FPEMsg#13
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL154
F1_000586 equ $ ; IN [MITFPGUN.BAS] CASE $23 : STR FPEMSG = "TIMED OUT.          ",0
        movlw 35
        subwf FPECode,W
        set@page BC@LL156
        btfss STATUS,2
        goto BC@LL156
ram_bank = 0
        movlw 84
        movwf FPEMsg#0
        movlw 73
        movwf FPEMsg#1
        movlw 77
        movwf FPEMsg#2
        movlw 69
        movwf FPEMsg#3
        movlw 68
        movwf FPEMsg#4
        movlw 32
```

```
        movwf FPEMsg#5
        movlw 79
        movwf FPEMsg#6
        movlw 85
        movwf FPEMsg#7
        movlw 84
        movwf FPEMsg#8
        movlw 46
        movwf FPEMsg#9
        movlw 32
        movwf FPEMsg#10
        movwf FPEMsg#11
        movwf FPEMsg#12
        movwf FPEMsg#13
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL156
F1_000587 equ $ ; IN [MITFPGUN.BAS] CASE $24 : STR FPEMSG = "UNAUTHORIZED.
",0
        movlw 36
        subwf FPECode,W
        set@page BC@LL158
        btfss STATUS,2
        goto BC@LL158
ram_bank = 0
        movlw 85
        movwf FPEMsg#0
        movlw 78
        movwf FPEMsg#1
        movlw 65
        movwf FPEMsg#2
        movlw 85
        movwf FPEMsg#3
        movlw 84
        movwf FPEMsg#4
        movlw 72
        movwf FPEMsg#5
        movlw 79
```

```
        movwf FPEMsg#6
        movlw 82
        movwf FPEMsg#7
        movlw 73
        movwf FPEMsg#8
        movlw 90
        movwf FPEMsg#9
        movlw 69
        movwf FPEMsg#10
        movlw 68
        movwf FPEMsg#11
        movlw 46
        movwf FPEMsg#12
        movlw 32
        movwf FPEMsg#13
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL158
F1_000588 equ $ ; IN [MITFPGUN.BAS] CASE $41 : STR FPEMSG = "COMMAND
CANCELLED.  ",0
        movlw 65
        subwf FPECode,W
        set@page BC@LL160
        btfss STATUS,2
        goto BC@LL160
ram_bank = 0
        movlw 67
        movwf FPEMsg#0
        movlw 79
        movwf FPEMsg#1
        movlw 77
        movwf FPEMsg#2
        movwf FPEMsg#3
        movlw 65
        movwf FPEMsg#4
        movlw 78
        movwf FPEMsg#5
        movlw 68
```

```
        movwf FPEMsg#6
        movlw 32
        movwf FPEMsg#7
        movlw 67
        movwf FPEMsg#8
        movlw 65
        movwf FPEMsg#9
        movlw 78
        movwf FPEMsg#10
        movlw 67
        movwf FPEMsg#11
        movlw 69
        movwf FPEMsg#12
        movlw 76
        movwf FPEMsg#13
        movwf FPEMsg#14
        movlw 69
        movwf FPEMsg#15
        movlw 68
        movwf FPEMsg#16
        movlw 46
        movwf FPEMsg#17
        movlw 32
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL160
F1_000589 equ $ ; IN [MITFPGUN.BAS] CASE $50 : STR FPEMSG = "INTERNAL ERROR.
",0
        movlw 80
        subwf FPECode,W
        set@page BC@LL162
        btfss STATUS,2
        goto BC@LL162
ram_bank = 0
        movlw 73
        movwf FPEMsg#0
        movlw 78
        movwf FPEMsg#1
        movlw 84
        movwf FPEMsg#2
        movlw 69
        movwf FPEMsg#3
```

```
        movlw 82
        movwf FPEMsg#4
        movlw 78
        movwf FPEMsg#5
        movlw 65
        movwf FPEMsg#6
        movlw 76
        movwf FPEMsg#7
        movlw 32
        movwf FPEMsg#8
        movlw 69
        movwf FPEMsg#9
        movlw 82
        movwf FPEMsg#10
        movwf FPEMsg#11
        movlw 79
        movwf FPEMsg#12
        movlw 82
        movwf FPEMsg#13
        movlw 46
        movwf FPEMsg#14
        movlw 32
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL162
F1_000590 equ $ ; IN [MITFPGUN.BAS] CASE $51 : STR FPEMSG = "MEMORY ERROR.
",0
        movlw 81
        subwf FPECode,W
        set@page BC@LL164
        btfss STATUS,2
        goto BC@LL164
ram_bank = 0
        movlw 77
        movwf FPEMsg#0
        movlw 69
        movwf FPEMsg#1
        movlw 77
        movwf FPEMsg#2
```

```
        movlw 79
        movwf FPEMsg#3
        movlw 82
        movwf FPEMsg#4
        movlw 89
        movwf FPEMsg#5
        movlw 32
        movwf FPEMsg#6
        movlw 69
        movwf FPEMsg#7
        movlw 82
        movwf FPEMsg#8
        movwf FPEMsg#9
        movlw 79
        movwf FPEMsg#10
        movlw 82
        movwf FPEMsg#11
        movlw 46
        movwf FPEMsg#12
        movlw 32
        movwf FPEMsg#13
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL164
F1_000591 equ $ ; IN [MITFPGUN.BAS] CASE $52 : STR FPEMSG = "FIRMWARE ERROR.
",0
        movlw 82
        subwf FPECode,W
        set@page BC@LL166
        btfss STATUS,2
        goto BC@LL166
ram_bank = 0
        movlw 70
        movwf FPEMsg#0
        movlw 73
        movwf FPEMsg#1
        movlw 82
        movwf FPEMsg#2
```

```
        movlw 77
        movwf FPEMsg#3
        movlw 87
        movwf FPEMsg#4
        movlw 65
        movwf FPEMsg#5
        movlw 82
        movwf FPEMsg#6
        movlw 69
        movwf FPEMsg#7
        movlw 32
        movwf FPEMsg#8
        movlw 69
        movwf FPEMsg#9
        movlw 82
        movwf FPEMsg#10
        movwf FPEMsg#11
        movlw 79
        movwf FPEMsg#12
        movlw 82
        movwf FPEMsg#13
        movlw 46
        movwf FPEMsg#14
        movlw 32
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL166
F1_000592 equ $ ; IN [MITFPGUN.BAS] CASE $60 : STR FPEMSG = "TEMPLATE ID
INVALID.",0
        movlw 96
        subwf FPECode,W
        set@page BC@LL168
        btfss STATUS,2
        goto BC@LL168
ram_bank = 0
        movlw 84
        movwf FPEMsg#0
        movlw 69
        movwf FPEMsg#1
```

```
        movlw 77
        movwf FPEMsg#2
        movlw 80
        movwf FPEMsg#3
        movlw 76
        movwf FPEMsg#4
        movlw 65
        movwf FPEMsg#5
        movlw 84
        movwf FPEMsg#6
        movlw 69
        movwf FPEMsg#7
        movlw 32
        movwf FPEMsg#8
        movlw 73
        movwf FPEMsg#9
        movlw 68
        movwf FPEMsg#10
        movlw 32
        movwf FPEMsg#11
        movlw 73
        movwf FPEMsg#12
        movlw 78
        movwf FPEMsg#13
        movlw 86
        movwf FPEMsg#14
        movlw 65
        movwf FPEMsg#15
        movlw 76
        movwf FPEMsg#16
        movlw 73
        movwf FPEMsg#17
        movlw 68
        movwf FPEMsg#18
        movlw 46
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL168
F1_000593 equ $ ; IN [MITFPGUN.BAS] CASE $61 : STR FPEMSG = "SECURIY LVL
INVALID.",0
        movlw 97
        subwf FPECode,W
        set@page BC@LL170
```

```
        btfss STATUS,2
        goto BC@LL170
ram_bank = 0
        movlw 83
        movwf FPEMsg#0
        movlw 69
        movwf FPEMsg#1
        movlw 67
        movwf FPEMsg#2
        movlw 85
        movwf FPEMsg#3
        movlw 82
        movwf FPEMsg#4
        movlw 73
        movwf FPEMsg#5
        movlw 89
        movwf FPEMsg#6
        movlw 32
        movwf FPEMsg#7
        movlw 76
        movwf FPEMsg#8
        movlw 86
        movwf FPEMsg#9
        movlw 76
        movwf FPEMsg#10
        movlw 32
        movwf FPEMsg#11
        movlw 73
        movwf FPEMsg#12
        movlw 78
        movwf FPEMsg#13
        movlw 86
        movwf FPEMsg#14
        movlw 65
        movwf FPEMsg#15
        movlw 76
        movwf FPEMsg#16
        movlw 73
        movwf FPEMsg#17
        movlw 68
        movwf FPEMsg#18
        movlw 46
        movwf FPEMsg#19
        clrf FPEMsg#20
```

```
        F@JUMP BC@LL136
BC@LL170
F1_000594 equ $ ; IN [MITFPGUN.BAS] CASE $62 : STR FPEMSG = "TIMEOUT VAL
INVALID.",0
        movlw 98
        subwf FPECode,W
        set@page BC@LL172
        btfss STATUS,2
        goto BC@LL172
ram_bank = 0
        movlw 84
        movwf FPEMsg#0
        movlw 73
        movwf FPEMsg#1
        movlw 77
        movwf FPEMsg#2
        movlw 69
        movwf FPEMsg#3
        movlw 79
        movwf FPEMsg#4
        movlw 85
        movwf FPEMsg#5
        movlw 84
        movwf FPEMsg#6
        movlw 32
        movwf FPEMsg#7
        movlw 86
        movwf FPEMsg#8
        movlw 65
        movwf FPEMsg#9
        movlw 76
        movwf FPEMsg#10
        movlw 32
        movwf FPEMsg#11
        movlw 73
        movwf FPEMsg#12
        movlw 78
        movwf FPEMsg#13
        movlw 86
        movwf FPEMsg#14
        movlw 65
        movwf FPEMsg#15
        movlw 76
        movwf FPEMsg#16
```

```
        movlw 73
        movwf FPEMsg#17
        movlw 68
        movwf FPEMsg#18
        movlw 46
        movwf FPEMsg#19
        clrf FPEMsg#20
        F@JUMP BC@LL136
BC@LL172
F1_000595 equ $ ; IN [MITFPGUN.BAS] CASE $63 : STR FPEMSG = "BAUD RATE INVALID.
",0
        movlw 99
        subwf FPECode,W
        set@page BC@LL174
        btfss STATUS,2
        goto BC@LL174
ram_bank = 0
        movlw 66
        movwf FPEMsg#0
        movlw 65
        movwf FPEMsg#1
        movlw 85
        movwf FPEMsg#2
        movlw 68
        movwf FPEMsg#3
        movlw 32
        movwf FPEMsg#4
        movlw 82
        movwf FPEMsg#5
        movlw 65
        movwf FPEMsg#6
        movlw 84
        movwf FPEMsg#7
        movlw 69
        movwf FPEMsg#8
        movlw 32
        movwf FPEMsg#9
        movlw 73
        movwf FPEMsg#10
        movlw 78
        movwf FPEMsg#11
        movlw 86
        movwf FPEMsg#12
        movlw 65
```

```
        movwf FPEMsg#13
        movlw 76
        movwf FPEMsg#14
        movlw 73
        movwf FPEMsg#15
        movlw 68
        movwf FPEMsg#16
        movlw 46
        movwf FPEMsg#17
        movlw 32
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
F1_000596 equ $ ; IN [MITFPGUN.BAS] CASE ELSE
        F@JUMP BC@LL176
BC@LL174
F1_000597 equ $ ; IN [MITFPGUN.BAS] STR FPEMSG = "UNKNOWN ERROR.      ",0
        movlw 85
        movwf FPEMsg#0
        movlw 78
        movwf FPEMsg#1
        movlw 75
        movwf FPEMsg#2
        movlw 78
        movwf FPEMsg#3
        movlw 79
        movwf FPEMsg#4
        movlw 87
        movwf FPEMsg#5
        movlw 78
        movwf FPEMsg#6
        movlw 32
        movwf FPEMsg#7
        movlw 69
        movwf FPEMsg#8
        movlw 82
        movwf FPEMsg#9
        movwf FPEMsg#10
        movlw 79
        movwf FPEMsg#11
        movlw 82
        movwf FPEMsg#12
        movlw 46
        movwf FPEMsg#13
```

```
        movlw 32
        movwf FPEMsg#14
        movwf FPEMsg#15
        movwf FPEMsg#16
        movwf FPEMsg#17
        movwf FPEMsg#18
        movwf FPEMsg#19
        clrf FPEMsg#20
F1_000599 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL176
BC@LL136
F1_000601 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_HWDiagnostics
F1_000605 equ $ ; IN [MITFPGUN.BAS] GLOCK = 0
        bcf PORTD,3
F1_000606 equ $ ; IN [MITFPGUN.BAS] GUNLOCK = 0
        bcf PORTD,2
sj_hwdStart
F1_000610 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000611 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"1.VERIFY"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB31)
        movwf GEN4H
        movlw low (STR@LB31)
        f@call C@STROUT
F1_000612 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"2.ENROLL"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB32)
        movwf GEN4H
        movlw low (STR@LB32)
        f@call C@STROUT
F1_000613 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"3.ERASE"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB33)
```

```
        movwf GEN4H
        movlw low (STR@LB33)
        f@call C@STROUT
F1_000614 equ $ ; IN [MITFPGUN.BAS] PRINT AT 4,1,"4.LOCK/UNLOCK"
        movlw 128
        movwf BPFH
        movlw 212
        f@call LCD@CRS
        movlw high (STR@LB34)
        movwf GEN4H
        movlw low (STR@LB34)
        f@call C@STROUT
F1_000616 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL177
F1_000618 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
ram_bank = 1
        clrf keytemp
F1_000619 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000621 equ $ ; IN [MITFPGUN.BAS] IF KEYTEMP = "1" THEN
        movlw 49
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL180
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL180
F1_000623 equ $ ; IN [MITFPGUN.BAS] FPC = 0
        clrf FPC
F1_000624 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCOMMAND
        f@call subs_FPCommand
F1_000626 equ $ ; IN [MITFPGUN.BAS] IF USERID <> 0 THEN
        bsf STATUS,5
ram_bank = 1
        movf userIDH,W
        iorwf userID,W
        set@page BC@LL182
        bcf STATUS,5
ram_bank = 0
```

```
        btfsc STATUS,2
        goto BC@LL182
F1_000628 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000629 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"USER: ", DEC USERID
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB35)
        movwf GEN4H
        movlw low (STR@LB35)
        f@call C@STROUT
        clrf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf userIDH,W
        bcf STATUS,5
ram_bank = 0
        movwf PP2H
        bsf STATUS,5
ram_bank = 1
        movf userID,W
        bcf STATUS,5
ram_bank = 0
        movwf PP2
        f@call OUT@DEC
        F@JUMP BC@LL183
BC@LL182
F1_000631 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000633 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000634 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"INVALID USER"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB36)
        movwf GEN4H
        movlw low (STR@LB36)
        f@call C@STROUT
F1_000636 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL183
F1_000638 equ $ ; IN [MITFPGUN.BAS] DELAYMS 3000
        movlw 11
        movwf PP1H
```

```
        movlw 184
        f@call DLY@W
F1_000640 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_HWDSTART
        F@JUMP sj_hwdStart
        F@JUMP BC@LL179
F1_000642 equ $ ; IN [MITFPGUN.BAS] ELSEIF KEYTEMP = "2" OR KEYTEMP = "3" THEN
BC@LL180
        movlw 50
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        movlw 1
        btfss STATUS,2
        movlw 0
        bcf STATUS,5
ram_bank = 0
        movwf SP#P9
        movlw 51
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        movlw 1
        btfss STATUS,2
        movlw 0
        bcf STATUS,5
ram_bank = 0
        iorwf SP#P9,F
        set@page BC@LL184
        btfsc STATUS,2
        goto BC@LL184
F1_000644 equ $ ; IN [MITFPGUN.BAS] FPC = KEYTEMP - $31
        movlw 49
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        bcf STATUS,5
ram_bank = 0
        movwf FPC
F1_000646 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000647 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"USER ID -> "
        movlw 128
        movwf BPFH
        f@call LCD@CRS
```

```
        movlw high (STR@LB37)
        movwf GEN4H
        movlw low (STR@LB37)
        f@call C@STROUT
F1_000649 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL185
F1_000651 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
ram_bank = 1
        clrf keytemp
F1_000652 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000653 equ $ ; IN [MITFPGUN.BAS] IF KEYTEMP <> "*" AND KEYTEMP <> "#" AND
KEYTEMP <> 0 THEN
        movlw 42
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        btfss STATUS,2
        movlw 1
        bcf STATUS,5
ram_bank = 0
        movwf SP#P9
        movlw 35
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        btfss STATUS,2
        movlw 1
        bcf STATUS,5
ram_bank = 0
        andwf SP#P9,F
        movlw 0
        bsf STATUS,5
ram_bank = 1
        movf keytemp,F
        btfss STATUS,2
        movlw 1
        bcf STATUS,5
ram_bank = 0
        andwf SP#P9,F
        set@page BC@LL188
```

```
        btfsc STATUS,2
        goto BC@LL188
F1_000656 equ $ ; IN [MITFPGUN.BAS] USERID = KEYTEMP - $30
        movlw 208
        bsf STATUS,5
ram_bank = 1
        addwf keytemp,W
        movwf userID
        movlw 255
        btfsc STATUS,0
        addlw 1
        movwf userIDH
F1_000657 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_FPCOMMAND
        bcf STATUS,5
ram_bank = 0
        f@call subs_FPCommand
F1_000659 equ $ ; IN [MITFPGUN.BAS] DELAYMS 3000
        movlw 11
        movwf PP1H
        movlw 184
        f@call DLY@W
F1_000661 equ $ ; IN [MITFPGUN.BAS] GOTO SJ_HWDSTART
        F@JUMP sj_hwdStart
F1_000663 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL188
F1_000665 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL185
BC@LL186
        F@JUMP BC@LL179
F1_000667 equ $ ; IN [MITFPGUN.BAS] ELSEIF KEYTEMP = "4" THEN
BC@LL184
        movlw 52
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL189
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL189
F1_000669 equ $ ; IN [MITFPGUN.BAS] GLOCK = ~GLOCK
        movlw 8
        xorwf PORTD,F
F1_000670 equ $ ; IN [MITFPGUN.BAS] GUNLOCK = ~GLOCK
```

```
        clrw
        btfss PORTD,3
        addlw 1
        btfsc STATUS,2
        bcf PORTD,2
        btfss STATUS,2
        bsf PORTD,2
F1_000672 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL189
BC@LL179
F1_000674 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL177
BC@LL178
F1_000678 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_LockGun
F1_000682 equ $ ; IN [MITFPGUN.BAS] GLOCK = 1
        bsf PORTD,3
F1_000683 equ $ ; IN [MITFPGUN.BAS] GUNLOCK = 0
        bcf PORTD,2
F1_000684 equ $ ; IN [MITFPGUN.BAS] DELAYMS 1000
        movlw 3
        movwf PP1H
        movlw 232
        f@call DLY@W
F1_000685 equ $ ; IN [MITFPGUN.BAS] GLOCK = 0
        bcf PORTD,3
F1_000686 equ $ ; IN [MITFPGUN.BAS] GUNLOCK = 0
        bcf PORTD,2
F1_000687 equ $ ; IN [MITFPGUN.BAS] DELAYMS 500
        movlw 1
        movwf PP1H
        movlw 244
        f@call DLY@W
F1_000689 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_UnLockGun
F1_000693 equ $ ; IN [MITFPGUN.BAS] GLOCK = 0
        bcf PORTD,3
F1_000694 equ $ ; IN [MITFPGUN.BAS] GUNLOCK = 1
        bsf PORTD,2
F1_000695 equ $ ; IN [MITFPGUN.BAS] DELAYMS 1000
        movlw 3
        movwf PP1H
```

```
        movlw 232
        f@call DLY@W
F1_000696 equ $ ; IN [MITFPGUN.BAS] GLOCK = 0
        bcf PORTD,3
F1_000697 equ $ ; IN [MITFPGUN.BAS] GUNLOCK = 0
        bcf PORTD,2
F1_000698 equ $ ; IN [MITFPGUN.BAS] DELAYMS 500
        movlw 1
        movwf PP1H
        movlw 244
        f@call DLY@W
F1_000700 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_SMS_Init
F1_000704 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000705 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"INITIALIZING"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB38)
        movwf GEN4H
        movlw low (STR@LB38)
        f@call C@STROUT
F1_000706 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"GSM MODULE...."
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB39)
        movwf GEN4H
        movlw low (STR@LB39)
        f@call C@STROUT
F1_000709 equ $ ; IN [MITFPGUN.BAS] GSMPW = 1
        bsf PORTC,5
F1_000710 equ $ ; IN [MITFPGUN.BAS] DELAYMS 2500
        movlw 9
        movwf PP1H
        movlw 196
        f@call DLY@W
F1_000711 equ $ ; IN [MITFPGUN.BAS] GSMPW = 0
        bcf PORTC,5
F1_000712 equ $ ; IN [MITFPGUN.BAS] DELAYMS 5000
        movlw 19
```

```
        movwf PP1H
        movlw 136
        f@call DLY@W
F1_000714 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT",13]
        movlw 65
        f@call HRSOUT
        movlw 84
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_000715 equ $ ; IN [MITFPGUN.BAS] DELAYMS 2000
        movlw 7
        movwf PP1H
        movlw 208
        f@call DLY@W
F1_000717 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"1"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw 49
        f@call PRINT
F1_000722 equ $ ; IN [MITFPGUN.BAS] HSERIN[WAIT("DST:")]
PB@LB191
        f@call HRSIN
        sublw 68
        set@page PB@LB191
        btfss STATUS,2
        goto PB@LB191
        f@call HRSIN
        sublw 83
        set@page PB@LB191
        btfss STATUS,2
        goto PB@LB191
        f@call HRSIN
        sublw 84
        set@page PB@LB191
        btfss STATUS,2
        goto PB@LB191
        f@call HRSIN
        sublw 58
        set@page PB@LB191
        btfss STATUS,2
        goto PB@LB191
```

```
F1_000723 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000725 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"2"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw 50
        f@call PRINT
F1_000727 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CIURC=0",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB40 >> 8) & 255
        movwf GEN4H
        movlw STR@LB40 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000728 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB192
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB192
        btfss STATUS,2
        goto PB@LB192
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB192
        btfss STATUS,2
        goto PB@LB192
F1_000729 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
```

```
        f@call DL@MS
F1_000731 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"3"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw 51
        f@call PRINT
F1_000733 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CNMI=3,1,0,0,0",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB41 >> 8) & 255
        movwf GEN4H
        movlw STR@LB41 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000734 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB193
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB193
        btfss STATUS,2
        goto PB@LB193
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB193
        btfss STATUS,2
        goto PB@LB193
F1_000735 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000737 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"4"
```

```
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw 52
        f@call PRINT
F1_000739 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CFUN=1",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB42 >> 8) & 255
        movwf GEN4H
        movlw STR@LB42 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000740 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB194
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB194
        btfss STATUS,2
        goto PB@LB194
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB194
        btfss STATUS,2
        goto PB@LB194
F1_000741 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000743 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"5"
        movlw 128
        movwf BPFH
```

```
        movlw 148
        f@call LCD@CRS
        movlw 53
        f@call PRINT
F1_000745 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CMGF=1",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB43 >> 8) & 255
        movwf GEN4H
        movlw STR@LB43 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000746 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB195
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB195
        btfss STATUS,2
        goto PB@LB195
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB195
        btfss STATUS,2
        goto PB@LB195
F1_000747 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000749 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"6"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
```

```
        movlw 54
        f@call PRINT
F1_000751 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CMEE=0",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB44 >> 8) & 255
        movwf GEN4H
        movlw STR@LB44 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000752 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB196
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB196
        btfss STATUS,2
        goto PB@LB196
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB196
        btfss STATUS,2
        goto PB@LB196
F1_000753 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000755 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"7"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw 55
        f@call PRINT
```

```
F1_000757 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CSDH=0",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB45 >> 8) & 255
        movwf GEN4H
        movlw STR@LB45 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000758 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB197
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB197
        btfss STATUS,2
        goto PB@LB197
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB197
        btfss STATUS,2
        goto PB@LB197
F1_000759 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000761 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"8"
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw 56
        f@call PRINT
F1_000763 equ $ ; IN [MITFPGUN.BAS] HSEROUT ["ATE1",13]
        movlw 65
```

```
        f@call HRSOUT
        movlw 84
        f@call HRSOUT
        movlw 69
        f@call HRSOUT
        movlw 49
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_000764 equ $ ; IN [MITFPGUN.BAS] HSERIN 1500,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 5
        movwf GENH
        movlw 220
        movwf GEN
PB@LB198
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB198
        btfss STATUS,2
        goto PB@LB198
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB198
        btfss STATUS,2
        goto PB@LB198
F1_000765 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_000767 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"STARTING IN "
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB46)
        movwf GEN4H
        movlw low (STR@LB46)
        f@call C@STROUT
```

```
F1_000769 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 10 TO 1 STEP -1
        movlw 10
        bsf STATUS,5
ram_bank = 1
        movwf gCtr
FR@LB199
        bcf STATUS,5
ram_bank = 0
        bsf STATUS,5
ram_bank = 1
        movf gCtr,F
        set@page NX@LB200
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,2
        goto NX@LB200
F1_000771 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,13, DEC2 GCTR, " SEC"
        movlw 128
        movwf BPFH
        movlw 160
        f@call LCD@CRS
        movlw 2
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
        movlw 32
        f@call PRINT
        movlw 83
        f@call PRINT
        movlw 69
        f@call PRINT
        movlw 67
        f@call PRINT
F1_000772 equ $ ; IN [MITFPGUN.BAS] DELAYMS 1000
        movlw 3
        movwf PP1H
        movlw 232
        f@call DLY@W
CT@LB201
F1_000774 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
```

```
        movlw 1
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,F
        set@page FR@LB199
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto FR@LB199
NX@LB200
F1_000776 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 0 TO 10
        bsf STATUS,5
ram_bank = 1
        clrf gCtr
FR@LB202
        bcf STATUS,5
ram_bank = 0
        movlw 11
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB203
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB203
F1_000778 equ $ ; IN [MITFPGUN.BAS] UNUM1[GCTR] = "0"
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw uNum1
        movwf FSR
        movlw 48
        movwf INDF
CT@LB204
        bcf STATUS,5
ram_bank = 0
F1_000780 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB202
        bcf STATUS,5
```

```
ram_bank = 0
        btfss STATUS,0
        goto FR@LB202
NX@LB203
F1_000782 equ $ ; IN [MITFPGUN.BAS] UNUM1[11] = 0
        bsf STATUS,5
ram_bank = 1
        clrf uNum1#11
F1_000786 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CPBF=",34,"ADMIN",34,13]
        movlw 4
        bcf STATUS,5
ram_bank = 0
        movwf BPFH
        movlw (STR@LB47 >> 8) & 255
        movwf GEN4H
        movlw STR@LB47 & 255
        f@call C@STROUT
        movlw 34
        f@call HRSOUT
        movlw 65
        f@call HRSOUT
        movlw 68
        f@call HRSOUT
        movlw 77
        f@call HRSOUT
        movlw 73
        f@call HRSOUT
        movlw 78
        f@call HRSOUT
        movlw 34
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_000787 equ $ ; IN [MITFPGUN.BAS] HSERIN 3500,
SUBS_SMS_INITFAILED,[WAIT("+CPBF:"),WAIT(34),STR UNUM1\11,WAIT("OK")]
        movlw 13
        movwf GENH
        movlw 172
        movwf GEN
PB@LB205
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
```

```
sublw 43
set@page PB@LB205
btfss STATUS,2
goto PB@LB205
f@call HRSIN2
set@page subs_SMS_InitFailed
btfss STATUS,0
goto subs_SMS_InitFailed
sublw 67
set@page PB@LB205
btfss STATUS,2
goto PB@LB205
f@call HRSIN2
set@page subs_SMS_InitFailed
btfss STATUS,0
goto subs_SMS_InitFailed
sublw 80
set@page PB@LB205
btfss STATUS,2
goto PB@LB205
f@call HRSIN2
set@page subs_SMS_InitFailed
btfss STATUS,0
goto subs_SMS_InitFailed
sublw 66
set@page PB@LB205
btfss STATUS,2
goto PB@LB205
f@call HRSIN2
set@page subs_SMS_InitFailed
btfss STATUS,0
goto subs_SMS_InitFailed
sublw 70
set@page PB@LB205
btfss STATUS,2
goto PB@LB205
f@call HRSIN2
set@page subs_SMS_InitFailed
btfss STATUS,0
goto subs_SMS_InitFailed
sublw 58
set@page PB@LB205
btfss STATUS,2
goto PB@LB205
```

```
PB@LB206
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 34
        set@page PB@LB206
        btfss STATUS,2
        goto PB@LB206
        clrf PBP#VAR0
PB@LB209
        bsf STATUS,0
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        movf PBP#VAR0,W
        addlw uNum1
        movwf FSR
        movf RCREG,W
        movwf INDF
        incf PBP#VAR0,F
        movf PBP#VAR0,W
        sublw 11
        set@page PB@LB209
        btfss STATUS,2
        goto PB@LB209
PB@LB210
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB210
        btfss STATUS,2
        goto PB@LB210
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB210
        btfss STATUS,2
        goto PB@LB210
```

```
F1_000788 equ $ ; IN [MITFPGUN.BAS] DELAYMS 1000
        movlw 3
        movwf PP1H
        movlw 232
        f@call DLY@W
F1_000794 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_SMS_SyncTime
sjumps_RetryGetClock
F1_000800 equ $ ; IN [MITFPGUN.BAS] TIMESTAMPS[20] = 0
        bsf STATUS,5
ram_bank = 1
        clrf timeStamps#20
F1_000802 equ $ ; IN [MITFPGUN.BAS] DELAYMS 500
        movlw 1
        bcf STATUS,5
ram_bank = 0
        movwf PP1H
        movlw 244
        f@call DLY@W
F1_000803 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CCLK?",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB48 >> 8) & 255
        movwf GEN4H
        movlw STR@LB48 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
F1_000804 equ $ ; IN [MITFPGUN.BAS] HSERIN 3000,
SJUMPS_RETRYGETCLOCK,[WAIT("+CCLK"),WAIT(34),STR TIMESTAMPS\20]
        movlw 11
        movwf GENH
        movlw 184
        movwf GEN
PB@LB211
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        sublw 43
        set@page PB@LB211
        btfss STATUS,2
        goto PB@LB211
```

```
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        sublw 67
        set@page PB@LB211
        btfss STATUS,2
        goto PB@LB211
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        sublw 67
        set@page PB@LB211
        btfss STATUS,2
        goto PB@LB211
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        sublw 76
        set@page PB@LB211
        btfss STATUS,2
        goto PB@LB211
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        sublw 75
        set@page PB@LB211
        btfss STATUS,2
        goto PB@LB211
PB@LB212
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        sublw 34
        set@page PB@LB212
        btfss STATUS,2
        goto PB@LB212
        clrf PBP#VAR0
PB@LB215
        bsf STATUS,0
```

```
        f@call HRSIN2
        set@page sjumps_RetryGetClock
        btfss STATUS,0
        goto sjumps_RetryGetClock
        movf PBP#VAR0,W
        addlw timeStamps
        movwf FSR
        movf RCREG,W
        movwf INDF
        incf PBP#VAR0,F
        movf PBP#VAR0,W
        sublw 20
        set@page PB@LB215
        btfss STATUS,2
        goto PB@LB215
F1_000805 equ $ ; IN [MITFPGUN.BAS] DELAYMS 500
        movlw 1
        movwf PP1H
        movlw 244
        f@call DLY@W
F1_000808 equ $ ; IN [MITFPGUN.BAS] TMPVAL[0] = TIMESTAMPS[0]
        bsf STATUS,5
ram_bank = 1
        movf timeStamps#0,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#0
F1_000809 equ $ ; IN [MITFPGUN.BAS] TMPVAL[1] = TIMESTAMPS[1]
        movf timeStamps#1,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#1
F1_000810 equ $ ; IN [MITFPGUN.BAS] TMPVAL[2] = 0
        clrf tmpVal#2
F1_000812 equ $ ; IN [MITFPGUN.BAS] YEARS = VAL TMPVAL,DEC
        movlw 218
```

```
        bcf STATUS,5
ram_bank = 0
        f@call VALDEC@32
        bsf STATUS,5
ram_bank = 1
        movwf years
        bcf STATUS,5
ram_bank = 0
        movf PP2H,W
        bsf STATUS,5
ram_bank = 1
        movwf yearsH
F1_000813 equ $ ; IN [MITFPGUN.BAS] YEARS = YEARS + 2000
        movlw 208
        addwf years,F
        movlw 7
        btfsc STATUS,0
        addlw 1
        addwf yearsH,F
F1_000815 equ $ ; IN [MITFPGUN.BAS] TMPVAL[0] = TIMESTAMPS[3]
        movf timeStamps#3,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#0
F1_000816 equ $ ; IN [MITFPGUN.BAS] TMPVAL[1] = TIMESTAMPS[4]
        movf timeStamps#4,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#1
F1_000817 equ $ ; IN [MITFPGUN.BAS] TMPVAL[2] = 0
        clrf tmpVal#2
F1_000819 equ $ ; IN [MITFPGUN.BAS] MONTHS = VAL TMPVAL,DEC
        movlw 218
        bcf STATUS,5
ram_bank = 0
        f@call VALDEC@32
```

```
        bsf STATUS,5
ram_bank = 1
        movwf months
F1_000821 equ $ ; IN [MITFPGUN.BAS] TMPVAL[0] = TIMESTAMPS[6]
        movf timeStamps#6,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#0
F1_000822 equ $ ; IN [MITFPGUN.BAS] TMPVAL[1] = TIMESTAMPS[7]
        movf timeStamps#7,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#1
F1_000823 equ $ ; IN [MITFPGUN.BAS] TMPVAL[2] = 0
        clrf tmpVal#2
F1_000825 equ $ ; IN [MITFPGUN.BAS] DAYS = VAL TMPVAL,DEC
        movlw 218
        bcf STATUS,5
ram_bank = 0
        f@call VALDEC@32
        bsf STATUS,5
ram_bank = 1
        movwf days
F1_000827 equ $ ; IN [MITFPGUN.BAS] TMPVAL[0] = TIMESTAMPS[9]
        movf timeStamps#9,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#0
F1_000828 equ $ ; IN [MITFPGUN.BAS] TMPVAL[1] = TIMESTAMPS[10]
        movf timeStamps#10,W
        bcf STATUS,5
ram_bank = 0
```

```
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#1
F1_000829 equ $ ; IN [MITFPGUN.BAS] TMPVAL[2] = 0
        clrf tmpVal#2
F1_000831 equ $ ; IN [MITFPGUN.BAS] HR = VAL TMPVAL, DEC
        movlw 218
        bcf STATUS,5
ram_bank = 0
        f@call VALDEC@32
        bsf STATUS,5
ram_bank = 1
        movwf hr
F1_000833 equ $ ; IN [MITFPGUN.BAS] TMPVAL[0] = TIMESTAMPS[12]
        movf timeStamps#12,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#0
F1_000834 equ $ ; IN [MITFPGUN.BAS] TMPVAL[1] = TIMESTAMPS[13]
        movf timeStamps#13,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#1
F1_000835 equ $ ; IN [MITFPGUN.BAS] TMPVAL[2] = 0
        clrf tmpVal#2
F1_000837 equ $ ; IN [MITFPGUN.BAS] MN = VAL TMPVAL, DEC
        movlw 218
        bcf STATUS,5
ram_bank = 0
        f@call VALDEC@32
        bsf STATUS,5
ram_bank = 1
        movwf mn
F1_000839 equ $ ; IN [MITFPGUN.BAS] TMPVAL[0] = TIMESTAMPS[15]
```

```
        movf timeStamps#15,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#0
F1_000840 equ $ ; IN [MITFPGUN.BAS] TMPVAL[1] = TIMESTAMPS[16]
        movf timeStamps#16,W
        bcf STATUS,5
ram_bank = 0
        movwf PBP#VAR0
        movf PBP#VAR0,W
        bsf STATUS,5
ram_bank = 1
        movwf tmpVal#1
F1_000841 equ $ ; IN [MITFPGUN.BAS] TMPVAL[2] = 0
        clrf tmpVal#2
F1_000843 equu $ ; IN [MITFPGUN.BAS] SC = VAL TMPVAL, DEC
        movlw 218
        bcf STATUS,5
ram_bank = 0
        f@call VALDEC@32
        bsf STATUS,5
ram_bank = 1
        movwf sc
F1_000845 equ $ ; IN [MITFPGUN.BAS] IF HR = 0 THEN
        movf hr,F
        set@page BC@LL216
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL216
F1_000847 equ $ ; IN [MITFPGUN.BAS] HR = 12
        movlw 12
        bsf STATUS,5
ram_bank = 1
        movwf hr
F1_000848 equ $ ; IN [MITFPGUN.BAS] AMPM = "A"
        movlw 65
        movwf ampm
        bcf STATUS,5
ram_bank = 0
```

```
        F@JUMP BC@LL215
F1_000850 equ $ ; IN [MITFPGUN.BAS] ELSEIF HR = 12 THEN
BC@LL216
        movlw 12
        bsf STATUS,5
ram_bank = 1
        subwf hr,W
        set@page BC@LL217
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL217
F1_000852 equ $ ; IN [MITFPGUN.BAS] HR = 12
        movlw 12
        bsf STATUS,5
ram_bank = 1
        movwf hr
F1_000853 equ $ ; IN [MITFPGUN.BAS] AMPM = "P"
        movlw 80
        movwf ampm
        bcf STATUS,5
ram_bank = 0
        F@JUMP BC@LL215
F1_000855 equ $ ; IN [MITFPGUN.BAS] ELSEIF HR > 12 THEN
BC@LL217
        movlw 13
        bsf STATUS,5
ram_bank = 1
        subwf hr,W
        set@page BC@LL218
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto BC@LL218
F1_000857 equ $ ; IN [MITFPGUN.BAS] HR = HR - 12
        movlw 12
        bsf STATUS,5
ram_bank = 1
        subwf hr,F
F1_000858 equ $ ; IN [MITFPGUN.BAS] AMPM = "P"
        movlw 80
        movwf ampm
        bcf STATUS,5
ram_bank = 0
```

```
        F@JUMP BC@LL219
BC@LL218
F1_000860 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000862 equ $ ; IN [MITFPGUN.BAS] AMPM = "A"
        movlw 65
        bsf STATUS,5
ram_bank = 1
        movwf ampm
F1_000865 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL219
        bcf STATUS,5
ram_bank = 0
BC@LL215
F1_000868 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_SMS_InitFailed
F1_000872 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000873 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"SYSTEM ERROR:"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB49)
        movwf GEN4H
        movlw low (STR@LB49)
        f@call C@STROUT
F1_000874 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"GSM MODULE IS NOT"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB50)
        movwf GEN4H
        movlw low (STR@LB50)
        f@call C@STROUT
F1_000875 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"RESPONDING. "
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB51)
        movwf GEN4H
        movlw low (STR@LB51)
        f@call C@STROUT
```

```
F1_000877 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_LOCKGUN
        f@call subs_LockGun
F1_000879 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL220
F1_000881 equ $ ; IN [MITFPGUN.BAS] MYBUZZER = ~MYBUZZER
        movlw 16
        xorwf PORTC,F
F1_000882 equ $ ; IN [MITFPGUN.BAS] DELAYMS 500
        movlw 1
        movwf PP1H
        movlw 244
        f@call DLY@W
F1_000885 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL220
BC@LL221
F1_000888 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_SMS_RegisterUser
sjumps_SRUStart
F1_000895 equ $ ; IN [MITFPGUN.BAS] FOR GCTR = 0 TO 10
        bsf STATUS,5
ram_bank = 1
        clrf gCtr
FR@LB223
        bcf STATUS,5
ram_bank = 0
        movlw 11
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page NX@LB224
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto NX@LB224
F1_000897 equ $ ; IN [MITFPGUN.BAS] UNUM1[GCTR] = "0"
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw uNum1
        movwf FSR
        movlw 48
        movwf INDF
CT@LB225
```

```
        bcf STATUS,5
ram_bank = 0
F1_000899 equ $ ; IN [MITFPGUN.BAS] NEXT GCTR
        movlw 1
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,F
        set@page FR@LB223
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,0
        goto FR@LB223
NX@LB224
F1_000901 equ $ ; IN [MITFPGUN.BAS] UNUM1[11] = 0
        bsf STATUS,5
ram_bank = 1
        clrf uNum1#11
F1_000902 equ $ ; IN [MITFPGUN.BAS] GCTR = 0
        clrf gCtr
sjumps_SRU_encodeNum
        bcf STATUS,5
ram_bank = 0
F1_000906 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000907 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"ENTER 11-DIGIT"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB52)
        movwf GEN4H
        movlw low (STR@LB52)
        f@call C@STROUT
F1_000908 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"DESTINATION NUMBER:"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB53)
        movwf GEN4H
        movlw low (STR@LB53)
        f@call C@STROUT
F1_000909 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"~ ",STR UNUM1\11
        movlw 128
        movwf BPFH
```

```
        movlw 148
        f@call LCD@CRS
        movlw 126
        f@call PRINT
        movlw 32
        f@call PRINT
        movlw 11
        movwf GEN4
        clrf PP2H
        movlw 185
        f@call SB@STROUT
F1_000911 equ $ ; IN [MITFPGUN.BAS] PRINT $FE, $0F
        movlw 254
        f@call PRINT
        movlw 15
        f@call PRINT
F1_000914 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL227
F1_000916 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3, GCTR + 3
        movlw 128
        movwf BPFH
        movlw 3
        movwf GEN
        movlw 3
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,W
        bcf STATUS,5
ram_bank = 0
        f@call CURSOR
F1_000918 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
ram_bank = 1
        clrf keytemp
F1_000919 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000921 equ $ ; IN [MITFPGUN.BAS] IF KEYTEMP = "#" THEN
        movlw 35
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL230
```

```
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL230
F1_000923 equ $ ; IN [MITFPGUN.BAS] IF GCTR < 10 THEN
        movlw 10
        bsf STATUS,5
ram_bank = 1
        subwf gCtr,W
        set@page BC@LL232
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,0
        goto BC@LL232
F1_000925 equ $ ; IN [MITFPGUN.BAS] INC GCTR
        bsf STATUS,5
ram_bank = 1
        incf gCtr,F
        bcf STATUS,5
ram_bank = 0
        F@JUMP BC@LL233
BC@LL232
F1_000927 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000929 equ $ ; IN [MITFPGUN.BAS] BREAK
        F@JUMP BC@LL228
F1_000931 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL233
        F@JUMP BC@LL229
F1_000933 equ $ ; IN [MITFPGUN.BAS] ELSEIF KEYTEMP = "*" THEN
BC@LL230
        movlw 42
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL234
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL234
F1_000935 equ $ ; IN [MITFPGUN.BAS] IF GCTR > 0 THEN
        bsf STATUS,5
ram_bank = 1
        movf gCtr,F
        set@page BC@LL236
```

```
        bcf STATUS,5
ram_bank = 0
        btfsc STATUS,2
        goto BC@LL236
F1_000937 equ $ ; IN [MITFPGUN.BAS] DEC GCTR
        bsf STATUS,5
ram_bank = 1
        decf gCtr,F
F1_000939 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL236
        bcf STATUS,5
ram_bank = 0
        F@JUMP BC@LL237
BC@LL234
F1_000941 equ $ ; IN [MITFPGUN.BAS] ELSE
F1_000943 equ $ ; IN [MITFPGUN.BAS] UNUM1[GCTR] = KEYTEMP
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw uNum1
        movwf FSR
        movf keytemp,W
        movwf INDF
F1_000945 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL237
        bcf STATUS,5
ram_bank = 0
BC@LL229
F1_000947 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3, GCTR + 3, UNUM1[GCTR]
        movlw 128
        movwf BPFH
        movlw 3
        movwf GEN
        movlw 3
        bsf STATUS,5
ram_bank = 1
        addwf gCtr,W
        bcf STATUS,5
ram_bank = 0
        f@call CURSOR
        bsf STATUS,5
ram_bank = 1
        movf gCtr,W
        addlw uNum1
```

```
        movwf FSR
        movf INDF,W
        bcf STATUS,5
ram_bank = 0
        f@call PRINT
F1_000950 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL227
BC@LL228
sjumps_SRU_confirmNum
F1_000954 equ $ ; IN [MITFPGUN.BAS] PRINT $FE, $0C
        movlw 254
        f@call PRINT
        movlw 12
        f@call PRINT
F1_000955 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,20,127
        movlw 128
        movwf BPFH
        movlw 167
        f@call LCD@CRS
        movlw 127
        f@call PRINT
F1_000957 equ $ ; IN [MITFPGUN.BAS] PRINT AT 4,1,"# - OK | * - CANCEL"
        movlw 128
        movwf BPFH
        movlw 212
        f@call LCD@CRS
        movlw high (STR@LB54)
        movwf GEN4H
        movlw low (STR@LB54)
        f@call C@STROUT
F1_000959 equ $ ; IN [MITFPGUN.BAS] WHILE 1 = 1
BC@LL238
F1_000961 equ $ ; IN [MITFPGUN.BAS] KEYTEMP = 0
        bsf STATUS,5
ram_bank = 1
        clrf keytemp
F1_000962 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_KP3X416FR
        bcf STATUS,5
ram_bank = 0
        f@call subs_KP3x416FR
F1_000964 equ $ ; IN [MITFPGUN.BAS] IF KEYTEMP = "#" THEN
        movlw 35
        bsf STATUS,5
ram_bank = 1
```

```
        subwf keytemp,W
        set@page BC@LL241
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL241
F1_000966 equ $ ; IN [MITFPGUN.BAS] BREAK
        F@JUMP BC@LL239
        F@JUMP BC@LL240
F1_000968 equ $ ; IN [MITFPGUN.BAS] ELSEIF KEYTEMP = "*" THEN
BC@LL241
        movlw 42
        bsf STATUS,5
ram_bank = 1
        subwf keytemp,W
        set@page BC@LL242
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL242
F1_000970 equ $ ; IN [MITFPGUN.BAS] GCTR = 10
        movlw 10
        bsf STATUS,5
ram_bank = 1
        movwf gCtr
F1_000971 equ $ ; IN [MITFPGUN.BAS] GOTO SJUMPS_SRU_ENCODENUM
        bcf STATUS,5
ram_bank = 0
        F@JUMP sjumps_SRU_encodeNum
F1_000974 equ $ ; IN [MITFPGUN.BAS] ENDIF
BC@LL242
BC@LL240
F1_000976 equ $ ; IN [MITFPGUN.BAS] WEND
        F@JUMP BC@LL238
BC@LL239
F1_000980 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000981 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"SAVING SMS"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB55)
        movwf GEN4H
        movlw low (STR@LB55)
```

```
        f@call C@STROUT
F1_000982 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"DESTINATION NUMBER"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB56)
        movwf GEN4H
        movlw low (STR@LB56)
        f@call C@STROUT
F1_000983 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"PLEASE WAIT.."
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB57)
        movwf GEN4H
        movlw low (STR@LB57)
        f@call C@STROUT
F1_000985 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CPBW=101,",34,STR
UNUM1\11,34,",129,",34,"ADMIN",34,13]
        movlw 4
        movwf BPFH
        movlw (STR@LB58 >> 8) & 255
        movwf GEN4H
        movlw STR@LB58 & 255
        f@call C@STROUT
        movlw 34
        f@call HRSOUT
        movlw 11
        movwf GEN4
        clrf PP2H
        movlw 185
        f@call SB@STROUT
        movlw 34
        f@call HRSOUT
        movlw 44
        f@call HRSOUT
        movlw 49
        f@call HRSOUT
        movlw 50
        f@call HRSOUT
        movlw 57
        f@call HRSOUT
```

```
        movlw 44
        f@call HRSOUT
        movlw 34
        f@call HRSOUT
        movlw 65
        f@call HRSOUT
        movlw 68
        f@call HRSOUT
        movlw 77
        f@call HRSOUT
        movlw 73
        f@call HRSOUT
        movlw 78
        f@call HRSOUT
        movlw 34
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_000986 equ $ ; IN [MITFPGUN.BAS] HSERIN 3000,
SUBS_SMS_INITFAILED,[WAIT("OK")]
        movlw 11
        movwf GENH
        movlw 184
        movwf GEN
PB@LB246
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB246
        btfss STATUS,2
        goto PB@LB246
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB246
        btfss STATUS,2
        goto PB@LB246
F1_000988 equ $ ; IN [MITFPGUN.BAS] DELAYMS 3000
        movlw 11
        movwf PP1H
```

```
        movlw 184
        f@call DLY@W
F1_000990 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_000991 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"SMS DESTINATION "
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB59)
        movwf GEN4H
        movlw low (STR@LB59)
        f@call C@STROUT
F1_000992 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"NUMBER SAVED."
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB60)
        movwf GEN4H
        movlw low (STR@LB60)
        f@call C@STROUT
F1_000994 equ $ ; IN [MITFPGUN.BAS] DELAYMS 3000
        movlw 11
        movwf PP1H
        movlw 184
        f@call DLY@W
F1_000996 equ $ ; IN [MITFPGUN.BAS] GOTO SJUMPS_SRU_END
        F@JUMP sjumps_SRU_End
sjumps_SRU_Fail
F1_001000 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_001001 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"SMS DESTINATION "
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB61)
        movwf GEN4H
        movlw low (STR@LB61)
        f@call C@STROUT
F1_001002 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"NUMBER REGISTRATION"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
```

```
        movlw high (STR@LB62)
        movwf GEN4H
        movlw low (STR@LB62)
        f@call C@STROUT
F1_001003 equ $ ; IN [MITFPGUN.BAS] PRINT AT 3,1,"CANCELLED."
        movlw 128
        movwf BPFH
        movlw 148
        f@call LCD@CRS
        movlw high (STR@LB63)
        movwf GEN4H
        movlw low (STR@LB63)
        f@call C@STROUT
F1_001005 equ $ ; IN [MITFPGUN.BAS] DELAYMS 3000
        movlw 11
        movwf PP1H
        movlw 184
        f@call DLY@W
sjumps_SRU_End
F1_001009 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
subs_SendData
F1_001013 equ $ ; IN [MITFPGUN.BAS] CLS
        f@call LCD@CLS
F1_001014 equ $ ; IN [MITFPGUN.BAS] PRINT AT 1,1,"SENDING DATA,"
        movlw 128
        movwf BPFH
        f@call LCD@CRS
        movlw high (STR@LB64)
        movwf GEN4H
        movlw low (STR@LB64)
        f@call C@STROUT
F1_001015 equ $ ; IN [MITFPGUN.BAS] PRINT AT 2,1,"PLEASE WAIT"
        movlw 128
        movwf BPFH
        movlw 192
        f@call LCD@CRS
        movlw high (STR@LB65)
        movwf GEN4H
        movlw low (STR@LB65)
        f@call C@STROUT
F1_001018 equ $ ; IN [MITFPGUN.BAS] GOSUB SUBS_SMS_SYNCTIME
        f@call subs_SMS_SyncTime
F1_001020 equ $ ; IN [MITFPGUN.BAS] HSEROUT["AT+CMGS=",34,STR UNUM1\11,34,13]
```

```
        movlw 4
        movwf BPFH
        movlw (STR@LB66 >> 8) & 255
        movwf GEN4H
        movlw STR@LB66 & 255
        f@call C@STROUT
        movlw 34
        f@call HRSOUT
        movlw 11
        movwf GEN4
        clrf PP2H
        movlw 185
        f@call SB@STROUT
        movlw 34
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_001021 equ $ ; IN [MITFPGUN.BAS] HSERIN 3500,
SUBS_SMS_INITFAILED,[WAIT(">")]
        movlw 13
        movwf GENH
        movlw 172
        movwf GEN
PB@LB249
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 62
        set@page PB@LB249
        btfss STATUS,2
        goto PB@LB249
F1_001022 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_001023 equ $ ; IN [MITFPGUN.BAS] HSEROUT["SMART GUN DATA REPORT",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB67 >> 8) & 255
        movwf GEN4H
        movlw STR@LB67 & 255
        f@call C@STROUT
        movlw 13
        f@call HRSOUT
```

```
F1_001024 equ $ ; IN [MITFPGUN.BAS] HSEROUT["USER LOGGED:"]
        movlw 4
        movwf BPFH
        movlw (STR@LB68 >> 8) & 255
        movwf GEN4H
        movlw STR@LB68 & 255
        f@call C@STROUT
F1_001026 equ $ ; IN [MITFPGUN.BAS] SELECT CASE USERID
F1_001028 equ $ ; IN [MITFPGUN.BAS] CASE 1
        bsf STATUS,5
ram_bank = 1
        decf userID,W
        iorwf userIDH,W
        set@page BC@LL250
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL250
F1_001030 equ $ ; IN [MITFPGUN.BAS] HSEROUT["ADMIN",13]
        movlw 65
        f@call HRSOUT
        movlw 68
        f@call HRSOUT
        movlw 77
        f@call HRSOUT
        movlw 73
        f@call HRSOUT
        movlw 78
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
        F@JUMP BC@LL249
BC@LL250
F1_001032 equ $ ; IN [MITFPGUN.BAS] CASE 2
        bsf STATUS,5
ram_bank = 1
        movf userID,W
        xorlw 2
        iorwf userIDH,W
        set@page BC@LL253
        bcf STATUS,5
ram_bank = 0
        btfss STATUS,2
        goto BC@LL253
```

```
F1_001034 equ $ ; IN [MITFPGUN.BAS] HSEROUT["USER",13]
        movlw 85
        f@call HRSOUT
        movlw 83
        f@call HRSOUT
        movlw 69
        f@call HRSOUT
        movlw 82
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_001036 equ $ ; IN [MITFPGUN.BAS] ENDSELECT
BC@LL253
BC@LL249
F1_001038 equ $ ; IN [MITFPGUN.BAS] HSEROUT["DATE: ",DEC2 MONTHS,"/",DEC2
DAYS,"/",DEC4 YEARS,13]
        movlw 4
        movwf BPFH
        movlw (STR@LB69 >> 8) & 255
        movwf GEN4H
        movlw STR@LB69 & 255
        f@call C@STROUT
        movlw 2
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf months,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
        movlw 47
        f@call HRSOUT
        movlw 2
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf days,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
        movlw 47
        f@call HRSOUT
        movlw 4
        movwf GEN4H
```

```
        bsf STATUS,5
ram_bank = 1
        movf yearsH,W
        bcf STATUS,5
ram_bank = 0
        movwf PP2H
        bsf STATUS,5
ram_bank = 1
        movf years,W
        bcf STATUS,5
ram_bank = 0
        movwf PP2
        f@call OUT@DEC
        movlw 13
        f@call HRSOUT
F1_001039 equ $ ; IN [MITFPGUN.BAS] HSEROUT["TIME: ",DEC2 HR,":",DEC2
MN,32,AMPM,"M",13]
        movlw 4
        movwf BPFH
        movlw (STR@LB70 >> 8) & 255
        movwf GEN4H
        movlw STR@LB70 & 255
        f@call C@STROUT
        movlw 2
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf hr,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
        movlw 58
        f@call HRSOUT
        movlw 2
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf mn,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
        movlw 32
        f@call HRSOUT
        bsf STATUS,5
```

```
ram_bank = 1
        movf ampm,W
        bcf STATUS,5
ram_bank = 0
        f@call HRSOUT
        movlw 77
        f@call HRSOUT
        movlw 13
        f@call HRSOUT
F1_001040 equ $ ; IN [MITFPGUN.BAS] HSEROUT["SHOTS FIRED:", DEC3 SHOTCTR]
        movlw 4
        movwf BPFH
        movlw (STR@LB71 >> 8) & 255
        movwf GEN4H
        movlw STR@LB71 & 255
        f@call C@STROUT
        movlw 3
        movwf GEN4H
        bsf STATUS,5
ram_bank = 1
        movf shotCtr,W
        bcf STATUS,5
ram_bank = 0
        f@call OUT@DECC
F1_001041 equ $ ; IN [MITFPGUN.BAS] DELAYMS 100
        movlw 100
        f@call DL@MS
F1_001042 equ $ ; IN [MITFPGUN.BAS] HSEROUT[26]
        movlw 26
        f@call HRSOUT
F1_001043 equ $ ; IN [MITFPGUN.BAS] HSERIN 15000,
SUBS_SMS_INITFAILED,[WAIT("+CMGS"),WAIT("OK")]
        movlw 58
        movwf GENH
        movlw 152
        movwf GEN
PB@LB255
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 43
        set@page PB@LB255
        btfss STATUS,2
```

```
        goto PB@LB255
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 67
        set@page PB@LB255
        btfss STATUS,2
        goto PB@LB255
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 77
        set@page PB@LB255
        btfss STATUS,2
        goto PB@LB255
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 71
        set@page PB@LB255
        btfss STATUS,2
        goto PB@LB255
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 83
        set@page PB@LB255
        btfss STATUS,2
        goto PB@LB255
PB@LB256
        f@call HRSIN2
        set@page subs_SMS_InitFailed
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 79
        set@page PB@LB256
        btfss STATUS,2
        goto PB@LB256
        f@call HRSIN2
        set@page subs_SMS_InitFailed
```

```
        btfss STATUS,0
        goto subs_SMS_InitFailed
        sublw 75
        set@page PB@LB256
        btfss STATUS,2
        goto PB@LB256
F1_001045 equ $ ; IN [MITFPGUN.BAS] RETURN
        return
F1_001050 equ $ ; IN [MITFPGUN.BAS] END
PB@LB257
        SLEEP
        F@JUMP PB@LB257
F1_EOF equ $ ; MITFPGUN.BAS
PB@LB258
        F@JUMP PB@LB258
STR@LB1
STR@LB15
        de 32,32,32,32,32,83,77,65,82,84,32,71,85,78,0
STR@LB2
        de 32,80,82,69,83,83,32,35,32,84,79,32,85,78,76,79,67,75,0
STR@LB3
        de 76,79,71,73,78,32,65,84,84,69,77,80,84,83,58,32,0
STR@LB4
        de 86,69,82,73,70,89,73,78,71,32,70,80,82,73,78,84,32,32,32,32,0
STR@LB5
        de 65,67,67,69,83,83,32,68,69,78,73,69,68,32,32,32,32,32,32,32,0
STR@LB6
        de 49,46,67,72,65,78,71,69,32,70,73,78,71,69,82,80,82,73,78,84,0
STR@LB7
        de 50,46,67,72,65,78,71,69,32,83,77,83,32,78,85,77,66,69,82,0
STR@LB8
        de 51,46,85,78,76,79,67,75,32,71,85,78,0
STR@LB9
        de 52,46,69,88,73,84,0
STR@LB10
        de 49,46,67,72,65,78,71,69,32,65,68,77,73,78,32,70,80,0
STR@LB11
        de 50,46,67,72,65,78,71,69,32,85,83,69,82,32,70,80,0
STR@LB12
        de 51,46,67,65,78,67,69,76,0
STR@LB13
        de 68,69,76,69,84,73,78,71,32,70,80,0
STR@LB14
STR@LB57
```

de 80,76,69,65,83,69,32,87,65,73,84,46,46,0
STR@LB16
de 32,32,80,82,69,83,83,32,35,32,84,79,32,76,79,67,75,0
STR@LB17
de 83,72,79,84,83,58,32,0
STR@LB18
de 71,85,78,32,73,83,32,76,79,67,75,69,68,46,32,32,32,32,32,32,0
STR@LB19
de 86,69,82,73,70,89,73,78,71,0
STR@LB20
de 70,73,78,71,69,82,80,82,73,78,84,0
STR@LB21
de 69,78,82,79,76,76,73,78,71,32,70,80,0
STR@LB22
de 84,65,75,73,78,71,32,83,65,77,80,76,69,35,32,49,0
STR@LB23
de 84,65,75,73,78,71,32,83,65,77,80,76,69,35,32,0
STR@LB24
de 76,73,70,84,32,70,73,78,71,69,82,32,32,32,32,32,32,32,32,32,0
STR@LB25
STR@LB27
de 83,65,77,80,76,73,78,71,32,69,82,82,79,82,32,32,32,32,32,32,0
STR@LB26
de 83,65,77,80,76,73,78,71,32,67,79,77,80,76,69,84,69,32,32,32,0
STR@LB28
de 68,69,76,69,84,73,78,71,46,46,0
STR@LB29
de 85,83,69,82,32,68,69,76,69,84,69,68,46,32,32,32,32,32,32,32,0
STR@LB30
de 79,80,69,82,65,84,73,79,78,32,69,82,82,79,82,32,32,32,32,32,0
STR@LB31
de 49,46,86,69,82,73,70,89,0
STR@LB32
de 50,46,69,78,82,79,76,76,0
STR@LB33
de 51,46,69,82,65,83,69,0
STR@LB34
de 52,46,76,79,67,75,47,85,78,76,79,67,75,0
STR@LB35
de 85,83,69,82,58,32,0
STR@LB36
de 73,78,86,65,76,73,68,32,85,83,69,82,0
STR@LB37
de 85,83,69,82,32,73,68,32,45,62,32,0

STR@LB38
de 73,78,73,84,73,65,76,73,90,73,78,71,0
STR@LB39
de 71,83,77,32,77,79,68,85,76,69,46,46,46,46,0
STR@LB40
de 65,84,43,67,73,85,82,67,61,48,0
STR@LB41
de 65,84,43,67,78,77,73,61,51,44,49,44,48,44,48,44,48,0
STR@LB42
de 65,84,43,67,70,85,78,61,49,0
STR@LB43
de 65,84,43,67,77,71,70,61,49,0
STR@LB44
de 65,84,43,67,77,69,69,61,48,0
STR@LB45
de 65,84,43,67,83,68,72,61,48,0
STR@LB46
de 83,84,65,82,84,73,78,71,32,73,78,32,0
STR@LB47
de 65,84,43,67,80,66,70,61,0
STR@LB48
de 65,84,43,67,67,76,75,63,0
STR@LB49
de 83,89,83,84,69,77,32,69,82,82,79,82,58,0
STR@LB50
de 71,83,77,32,77,79,68,85,76,69,32,73,83,32,78,79,84,0
STR@LB51
de 82,69,83,80,79,78,68,73,78,71,46,32,0
STR@LB52
de 69,78,84,69,82,32,49,49,45,68,73,71,73,84,0
STR@LB53
de 68,69,83,84,73,78,65,84,73,79,78,32,78,85,77,66,69,82,58,0
STR@LB54
de 35,32,45,32,79,75,32,124,32,42,32,45,32,67,65,78,67,69,76,0
STR@LB55
de 83,65,86,73,78,71,32,83,77,83,0
STR@LB56
de 68,69,83,84,73,78,65,84,73,79,78,32,78,85,77,66,69,82,0
STR@LB58
de 65,84,43,67,80,66,87,61,49,48,49,44,0
STR@LB59
STR@LB61
de 83,77,83,32,68,69,83,84,73,78,65,84,73,79,78,32,0
STR@LB60

```
        de 78,85,77,66,69,82,32,83,65,86,69,68,46,0
STR@LB62
        de 78,85,77,66,69,82,32,82,69,71,73,83,84,82,65,84,73,79,78,0
STR@LB63
        de 67,65,78,67,69,76,76,69,68,46,0
STR@LB64
        de 83,69,78,68,73,78,71,32,68,65,84,65,44,0
STR@LB65
        de 80,76,69,65,83,69,32,87,65,73,84,0
STR@LB66
        de 65,84,43,67,77,71,83,61,0
STR@LB67
        de 83,77,65,82,84,32,71,85,78,32,68,65,84,65,32,82,69,80,79,82,84,0
STR@LB68
        de 85,83,69,82,32,76,79,71,71,69,68,58,0
STR@LB69
        de 68,65,84,69,58,32,0
STR@LB70
        de 84,73,77,69,58,32,0
STR@LB71
        de 83,72,79,84,83,32,70,73,82,69,68,58,0
__EOF
__config HS_OSC & WDT_OFF & PWRTE_ON & BODEN_OFF & LVP_OFF & CP_OFF &
CPD_OFF & DEBUG_OFF
        End
```

ADST11SD300&310

Fingerprint Identification Module

User's Guide

(Ver 4.0)

IDSystemTechnologyCo.,Ltd

http://www.szadst.com

Tel:86-755-33160920

Fax:86-755-86039586

List

1. SUMMARY 207
1.1 MAIN FUNCTIONS 207
1.2 APPLICATION EXAMPLES 208
1.3 TECHNOLOGY PARAMETER 208
1.4 DEFAULT SETTING 209
2. CONFIGRATION AND CONNECTOR 210
2.1 MOUNT AND CONNECTOR 210
2.2 DIMENSION OF ADST11SD300 211
2.3 DIMENSION OF ADST11SD310 211
3. COMMUNICATION PROTOCOL 212
3.1 THE PROCESS OF COMMUNICATION 212
3.2 CLASSIFY OF COMMUNICATION PACKET 213
3.2.1 Command packet 213
3.2.2 Response packet 213
3.2.3 Data Packet 213
3.3 PACKET STRUCTURE 213
3.3.1 Packet Identify Code 213
3.3.2 Structure of Command Packet 213
3.3.3 Response packet 214
3.3.4 Command Data Packet 215
3.3.5 Response Data Packet 215
4. PACKET COMMAND DETAIL DESCRIPTION 216
4.1 STRUCTURE OF FINGERPRINT TEMPLATE DATA IS FOLLOW 216
4.2 COMMAND LIST 216
5. APPENDIX 219
5.1 RESPONSE AND ERROR CODE 219
5.2 FLOW OF ENROLL AND VERIFY AND IDENTIFY PROCESS 221
5.2.1 Enroll 221
5.2.2 Verify & Identify 223

1. Summary

Functions and Characteristic:

ADST11SD3xx serial is All-in-one (Optic sensor +CMOS image sensor + CPU +Flash memory) embeddedstand-alone fingerprint identification module, it is convenient for second development,Extremely easy to integrate minimizing time-to-market.

1.1 Main functions

Optic sensor is reliable and Low-cost, High ESD Protection

1:N Identification (One-to-Many)

1:1 Verification (One-to-One)

High speed fingerprint identification algorithm engine

Self study function

Fingerprint template data read from /write to FLASH memory function

Get Feature Data of Captured fingerprint

Verify Downloaded Feature with Captured fingerprint

Identify Downloaded Feature with Captured fingerprint

Security Level setting

Standby mode

Communication interface :UART

1.2 Application examples

- Access control systems
- Time & Attendance
- Locks, safes
- POS terminals

1.3 Technology Parameter

Item	Description
CPU	Core:ARC (120Mhz,1M bit Flash Memory,128K SRAM)
Sensor	Optic sensor
Fingerprint capacity	500~5000 can be enlarge，Default setting is 3000 fingerprints
FAR	< 0.001 %（Security Level is 3）
FRR	< 0.1 %（Security Level is 3）
Match Mode	1:N identification and 1:1 verification
Fingerprint Data size	496 Byte
Security level	Level 1 to Level 5 can be set , default setting is : Level 3
Time of Response	Fingerprint pretreatment < 0.45

	1:N matching（3000 fingerprints）< 1s
Resistant to Abrasion	>1 Million Finger Press
Communication	UART：Parity = NONE,Stop Bit = 1,Flow Control = NONE Baud rate is 9600/19200/38400/57600/115200 (BPS) Default setting：115200bps
Working Voltage	DC 3.3V-5V
Working Current	Max 100mA，Standby Mode less than 5mA
Working Environment	Working Temperature：-10℃ to 60℃ Relative humidity：20%– 80%
Storage Environment	Storage temperature: -20℃ to 80℃ Storage humidity:<90%

1.4 Default Setting

Item	Value
Security Level（1～5）	3
Finger Time Out（0～60S）	5s
BaudRate（9600，19200，38400，57600，115200）BPS	115200 BPS
Fingerprint Duplication Check（ON/OFF）	ON

2. Configration and Connector

2.1 Mount and Connector

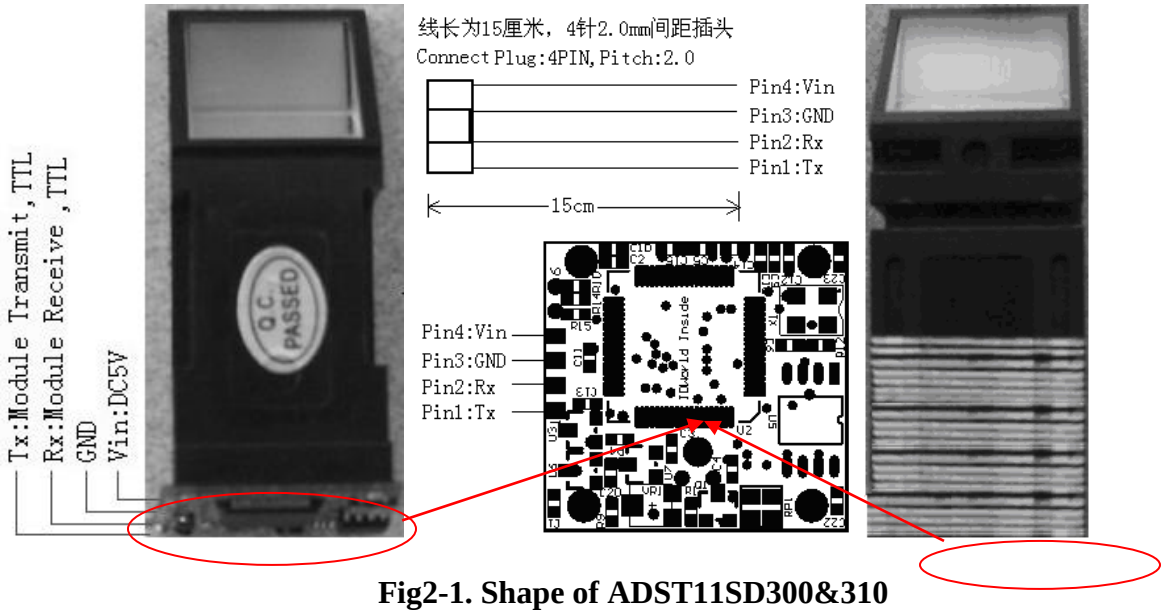


Fig2-1. Shape of ADST11SD300&310

Table2-1. Connector Signal description

Pin No.	Color	Signal name	Signal Description
PIN 1	White	UART-TX	Module Transmit Output,3.3V TTL Logic
PIN 2	Yellow	UART-RX	Module Receive Input,3.3V TTL Logic
PIN 3	Black	GND	GND
PIN 4	Red	VIN	DC Input,Range of voltage is 3.3V-5V

2.2 Dimension of ADST11SD300

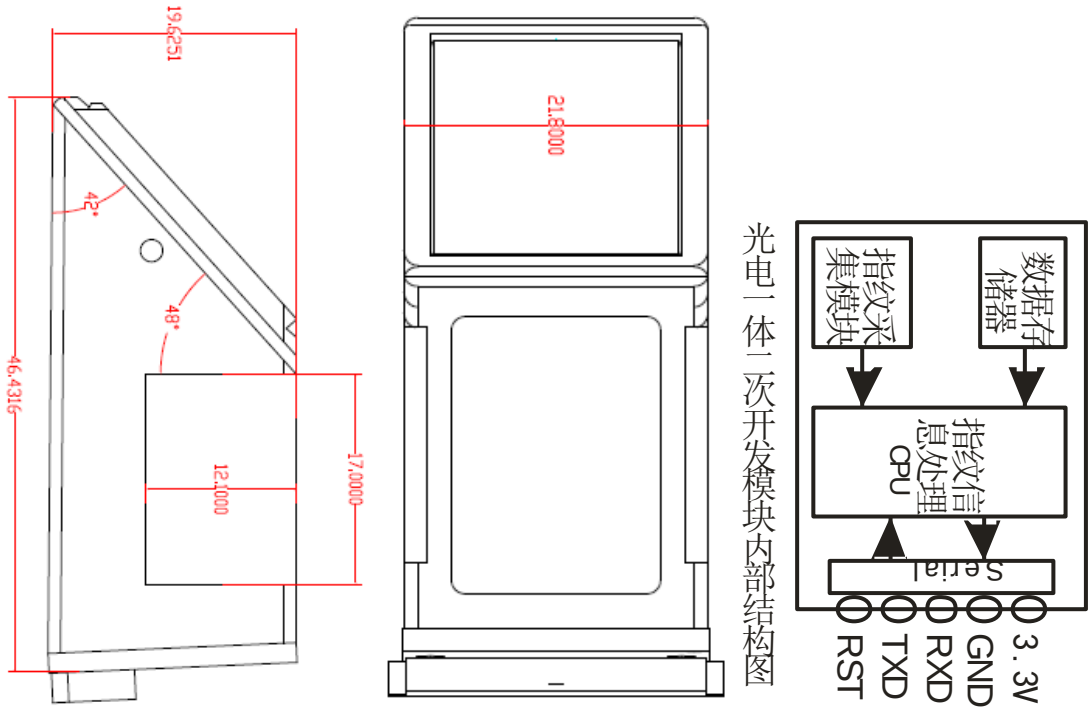


Fig2-2. Dimension of ADST11SD300

2.3 Dimension of ADST11SD310

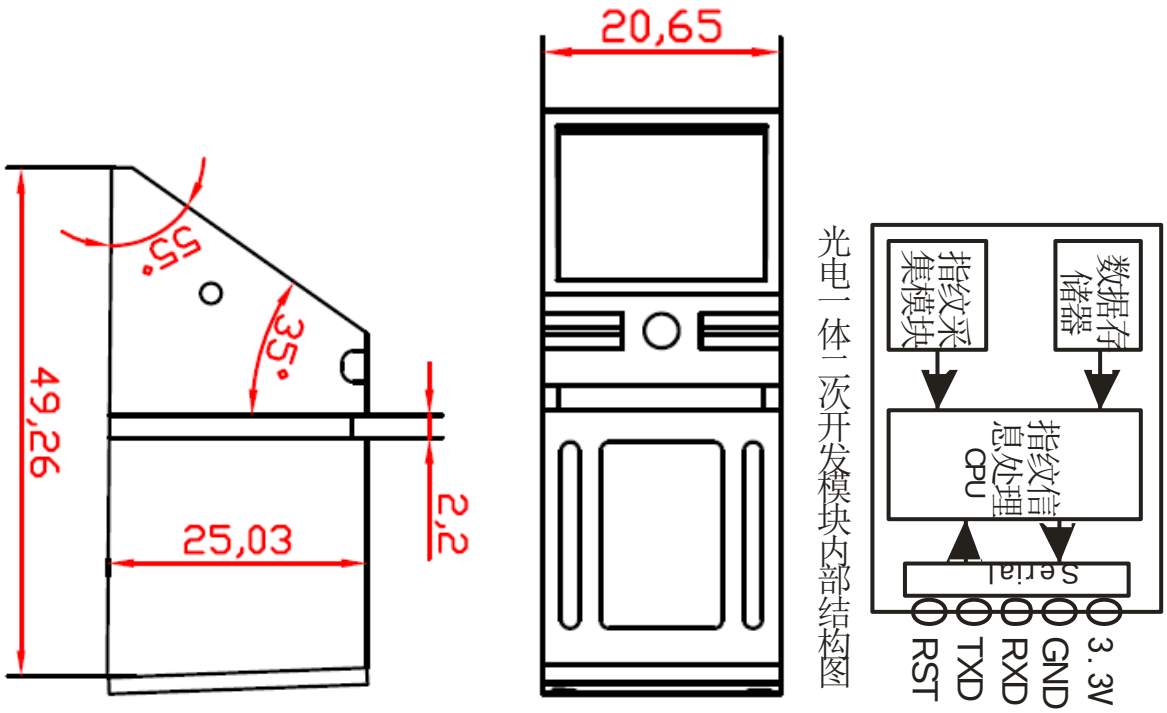


Fig 2-3. Dimension of ADST11SD310

3. Communication Protocol

The module is used as a slave device. The Master device sends relative commands to control it.

The CMD sent by the master and the ACK signal returned by the module.

Command interface: UART (Universal Asynchronous Receiver Transmitter)

Default Setting is : 115200BPS ,1 start-bit ,1 stop-bit ,no check bit

3.1 The process of communication

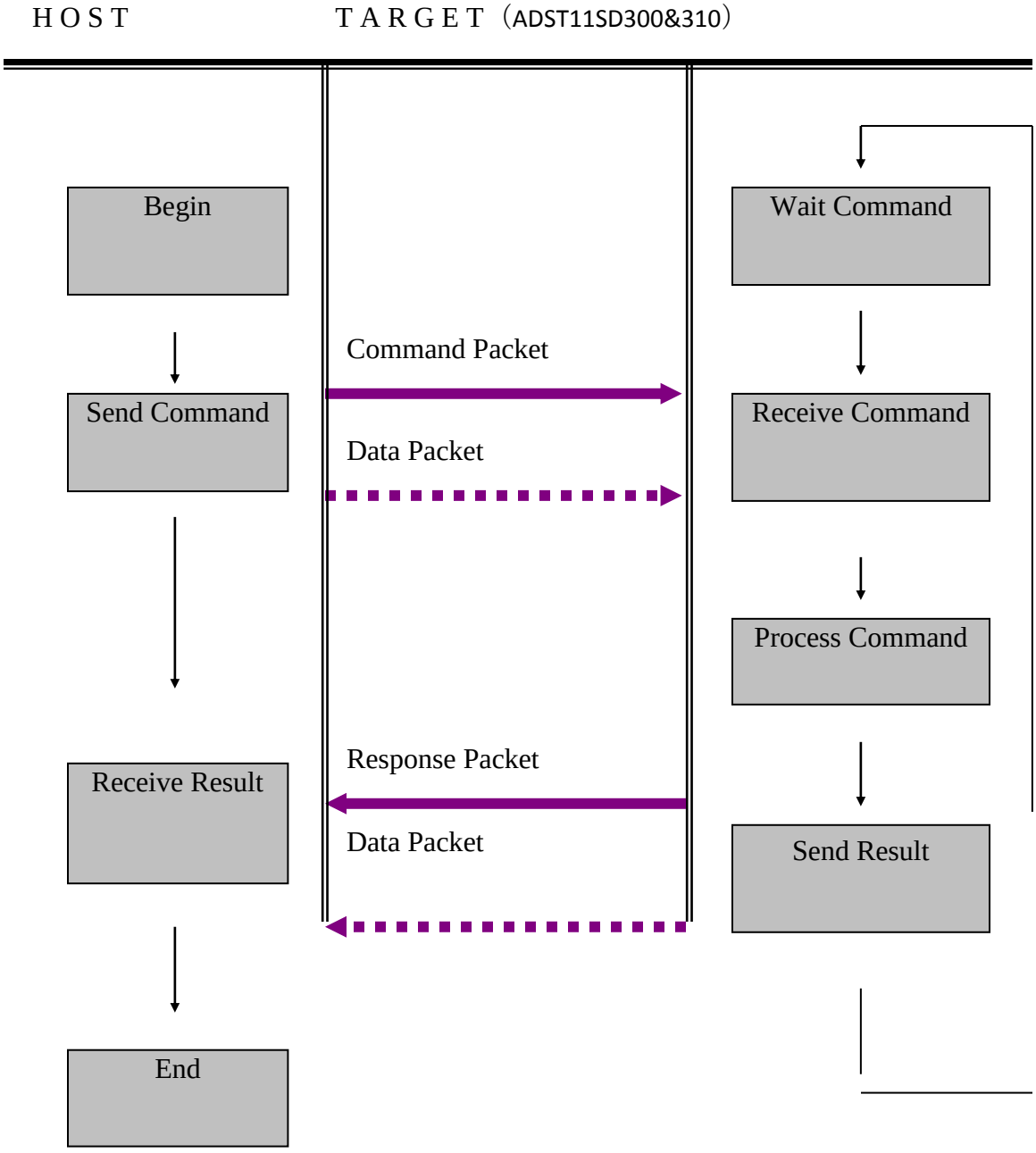


Fig3-1.Process of Communication

Notice. Host do not send next command until receive result , except send FP Cancel

3.2 Classify of communication packet

3.2.1 Command packet

Command Packet is the instruction from Host to Target (ADST11SD3xx),

Total length of the command packet is 24 Bytes

3.2.2 Response packet

Response packet is result of execute command packet, from Target (ADST11SD3xx) to Host,

Total length of the command packet is 24 Bytes

3.2.3 Data Packet

When length of Command Parameter or Data is larger than 16 Bytes, Utilize Data Packet

to transmit block Data, the maximum length of Data Packet is 512Bytes

3.3 Packet Structure

3.3.1 Packet Identify Code

Section of start 2byte prefix define type of packet

Type of Packet	Code
Command packet	0xAA55
Response packet	0x55AA
Command Data Packet	0xA55A
Response Data Packet	0x5AA5

3.3.2 Structure of Command Packet

PREFIX		CMD		LEN		DATA				CKS	
0x55	0xAA	L	H	L	H	D0	D1	...	D15	L	H

0	1	2	3	4	5	6	7	...	21	22	23

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	CMD	WORD	2byte	Command Code
4	LEN	WORD	2byte (=n, n < 17)	Length of Command Parameter
6	DATA	Byte array	16byte	Command Parameter
22	CKS	WORD	2byte	Check Sum is the low word of value listed below: offset[0] + offset[1] + ...+offset[21]

3.3.3 Response packet

PREFIX		RCM		LEN		RET		DATA				CKS	
0xAA	0x55	L	H	L	H	L	H	D0	D1	...	D13	L	H
0	1	2	3	4	5	6	7	8	9	...	21	22	23

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	RCM	WORD	2byte	Response Code
4	LEN	WORD	2byte(=n, n < 17)	RET and DATA
6	RET	WORD	2byte	Result Code (0 :success, 1 :fail)
8	DATA	Byte array	14byte	Response Data (n-2 byte)
22	CKS	WORD	2byte	Check Sum is the low word of value

				listed below: offset[0] + offset[1] + ...+offset[21]

3.3.4 Command Data Packet

PREFIX		CMD		LEN		DATA				CKS	
0x5A	0xA5	L	H	L	H	D0	D1	...	Dn-1	L	H
0	1	2	3	4	5	6	7	...	6+n-1	6+n	6+n+1

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code
2	CMD	WORD	2byte	Command Code
4	LEN	WORD	2byte(=n , n < 512)	Length of DATA
6	DATA	Byte array	nbyte	Command parameter
6+n	CKS	WORD	2byte	Check Sum is the low word of value listed below: offset[0] + offset[1] + ...+offset[6+n-1]

Before send Command Data packet, Host first send Command packet which set the length of next command data packet in Data Field

3.3.5 Response Data Packet

PREFIX		RCM		LEN		RET		DATA				CKS	
0xA5	0x5A	L	H	L	H	L	H	D0	D1	...	Dn-3	L	H
0	1	2	3	4	5	6	7	8	9	...	6+n-1	6+n	6+n+1

OFFSET	FIELD	TYPE	SIZE	DESCRYPTION
0	PREFIX	WORD	2byte	Packet Identify code

2	RCM	WORD	2byte	Response Code
4	LEN	WORD	2byte(=n, n < 512)	Length of result data：(RET + DATA)
6	RET	WORD	2byte	Result code（0 : success 1 : fail）
8	DATA	Byte array	（n-2）byte	Response data
6+n	CKS	WORD	2byte	Check Sum is the low word of value listed below: offset[0] + offset[1] + ...+offset[6+n-1]

4. Packet Command Detail Description

4.1 Structure of Fingerprint Template Data is follow

Template Data(496Bytes)+CheckSum(2Bytes)=498Bytes

Template Data	CheckSum
496 bytes Template Data D0,D1,...D495	2 byte Check Sum is the low word of value listed below: D0+D1+...+D495

4.2 Command List

No	Function	Command			Response			
		CMD	LEN	DATA	RCM	LEN	RET	DATA
1	Verify	0x0101	2	Template No.	0x0101	4	0/1	TemplateNo./ Error Code

No	Function	Command			Response			
2	Identify	0x0102	0	-	0x0102	4	0/1	Template No ./ Error Code
3	Enroll	0x0103	2	Template No.	0x0103	4/6	0/1	Template No. / Error Code + Duplicaion ID
4	Enroll One Time	0x0104	2	Template No.	0x0104	4/6	0/1	Template No. / ErrorCode
5	Clear Template	0x0105	2	Template No.	0x0105	4	0/1	Template No. / ErrorCode
6	Clear All Template	0x0106	0	-	0x0106	4	0/1	Cleared TemplateCount / Error Code
7	Get Empty ID	0x0107	0	-	0x0107	4	0/1	Template No. / Error Code
8	Get Template Status	0x0108	2	Template No.	0x0108	4	0/1	Template Status /Error Code
9	Get Broken Template	0x0109	0	-	0x0109	6	0/1	Broken Template Count + First Broken Template No./Error Code
10	Read Template	0x010A	2	Template No.	0x010A	(Template Record Size + 4) or (4)	0/1	Template No + Template Record Data /Error Code
11	Write Template	0x010B	Template Record Size + 2	Template No. + Template Record Data	0x010B	4	0/1	Template No. / Error Code
12	Set Security Level	0x010C	2	Security Level Value	0x010C	4	0/1	Security Level Value/Error Code

No	Function	Command			Response			
13	Get Security Level	0x010D	0	-	0x010D	4	0	Security Level Value
14	Set Finger Time Out	0x010E	2	Time Out Value	0x010E	4	0/1	TimeOut Value / Error Code
15	Get Finger Time Out	0x010F	0	-	0x010F	4	0	TimeOut Value
16	Set Device ID	0x0110	2	Device ID	0x0110	4	0	Device ID
17	Get Device ID	0x0111	0	-	0x0111	4	0/1	Device ID / Error Code
18	Get F/W Version	0x0112	0	-	0x0112	4	0	F/W Version
19	Finger Detect	0x0113	0	-	0x0113	4	0	Detect Result
20	Set BaudRate	0x0114	2	BaudRate Index	0x0114	4	0/1	BaudRate Index / Error Code
21	Set Duplication Check	0x0115	2	Duplication Check Option(1/0)	0x0115	4	0/1	Duplication Check Option /Error Code
22	Get Duplication Check	0x0116	0	-	0x0116	4	0	Duplication Check Option
23	Enter Stadby Mode	0x0117	0	-	0x0117	4	0	-
24	Enroll And Store in RAM	0x0118	0	-	0x0118	4	0/1	0 / ErrorCode
25	Get Enroll Data	0x0119	0	-	0x0119	(Template Record Size + 2) or (4)	0/1	Template Record Data /Error Code
26	Get Feature Data of Captured FP	0x011A	0	-	0x011A	(Template Record Size + 2) or (4)	0/1	Template Record Data /Error Code

No	Function	Command			Response			
27	Verify Downloaded Feature with Captured FP	0x011B	Template Record Size	Template Record Data	0x011B	4	0/1	0/Error Code
28	Identify Downloaded Feature with Captured FP	0x011C	Template Record Size + 2	Index + Template Record Data	0x011C	4	0/1	0/Error Code
29	Get Device Name	0x0121	0	-	0x0121	16	0/1	"STO20-OEM" /Error Code
30	Sensor LED Control	0x0124	2	0/1	0x0124	4	0	0
31	Identify Free	0x0125	0	-	0x0125	4	0/1	Template No. / Error Code
32	Set Device Password	0x0126	14	Password	0x0126	4	0/1	0/ Error Code
33	Verify Device Password	0x0127	14	Password	0x0127	4	0/1	0/ Error Code
34	Get Enroll Count	0x0128	0		0x0128	4	0/1	Enroll Count /Error Code
35	FP Cancel	0x0130	0	-	0x0130	2	0	-
36	Test Connection	0x0150	0	-	0x0150	2	0	-
37	Incorrect Command	-	-	-	0x0160	2	0	-

Table4-2 Command List

5. Appendix

5.1 Response and Error Code

No.	Response and Error Code	Value	Description
1	ERR_SUCCESS	0x00	Successful for command execute

2	ERR_FAIL	0x01	Fail for command execute
3	ERR_VERIFY	0x11	One to one match fail
4	ERR_IDENTIFY	0x12	Identify fail, no matched template
5	ERR_TMPL_EMPTY	0x13	The appointed Template Data is Null
6	ERR_TMPL_NOT_EMPTY	0x14	Existed Template Data for the appointed ID
7	ERR_ALL_TMPL_EMPTY	0x15	All template is Null, Without enrolled Template
8	ERR_EMPTY_ID_NOEXIST	0x16	Without available Template ID.
9	ERR_BROKEN_ID_NOEXIST	0x17	Without damaged Template.
10	ERR_INVALID_TMPL_DATA	0x18	Appointed Template Data is invalid.
11	ERR_DUPLICATION_ID	0x19	The fingerprint has been enrolled.
12	ERR_BAD_QUALITY	0x21	Bad quality fingerprint image.
13	ERR_TIME_OUT	0x23	During Time Out period, no finger is detected
14	ERR_NOT_AUTHORIZED	0x24	Command is without authorization
15	ERR_GENERALIZE	0x30	Generalize Template Data fail
16	ERR_FP_CANCEL	0x41	Relative fingerprint command have been cancelled
17	ERR_INTERNAL	0x50	Internal Error
18	ERR_MEMORY	0x51	Memory Error
19	ERR_EXCEPTION	0x52	Firmware Error
20	ERR_INVALID_TMPL_NO	0x60	Template No. is invalid
21	ERR_INVALID_SEC_VAL	0x61	The Value of Security Level is invalid
22	ERR_INVALID_TIME_OUT	0x62	The Value of Time Out is invalid
23	ERR_INVALID_BAUDRATE	0x63	The Value of Baud Rate is invalid
24	ERR_DEVICE_ID_EMPTY	0x64	Not setting Device ID
25	ERR_INVALID_ DUP_VAL	0x65	Option Value of Duplication Check is invalid

26	ERR_INVALID_PARAM	0x70	Parameter is invalid
27	ERR_NO_RELEASE	0x71	In process of IdentifyFree command, Finger is not release
28	GD_DOWNLOAD_SUCCESS	0xA1	Download Template Record Data is successful
29	GD_NEED_FIRST_SWEEP	0xFFF1	Waiting input fingerprint for the first time.
30	GD_NEED_SECOND_SWEEP	0xFFF2	Waiting input fingerprint for the second time.
31	GD_NEED_THIRD_SWEEP	0xFFF3	Waiting input fingerprint for the third time.
32	GD_NEED_RELEASE_FINGER	0xFFF4	Lift finger
33	GD_DETECT_FINGER	0x01	There is an finger on sensor
34	GD_NO_DETECT_FINGER	0x00	There is no finger on sensor
35	GD_TEMPLATE_NOT_EMPTY	0x01	The appointed Template Data are not null
36	GD_TEMPLATE_EMPTY	0x00	The appointed Template Data are null

5.2 Flow of Enroll and Verify and Identify process

5.2.1 Enroll

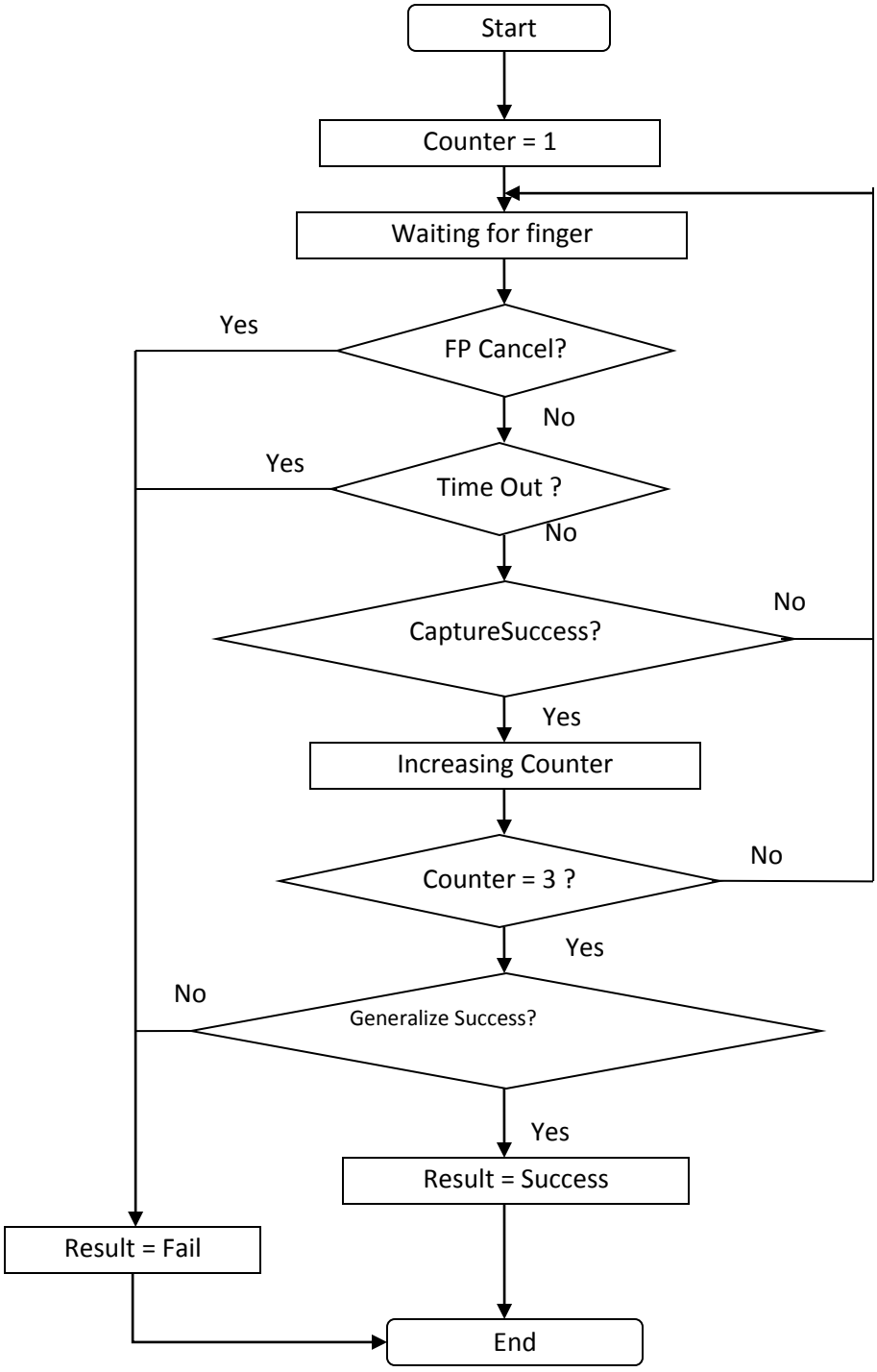


Fig 5-1 Flow of Enroll

5.2.2 Verify & Identify

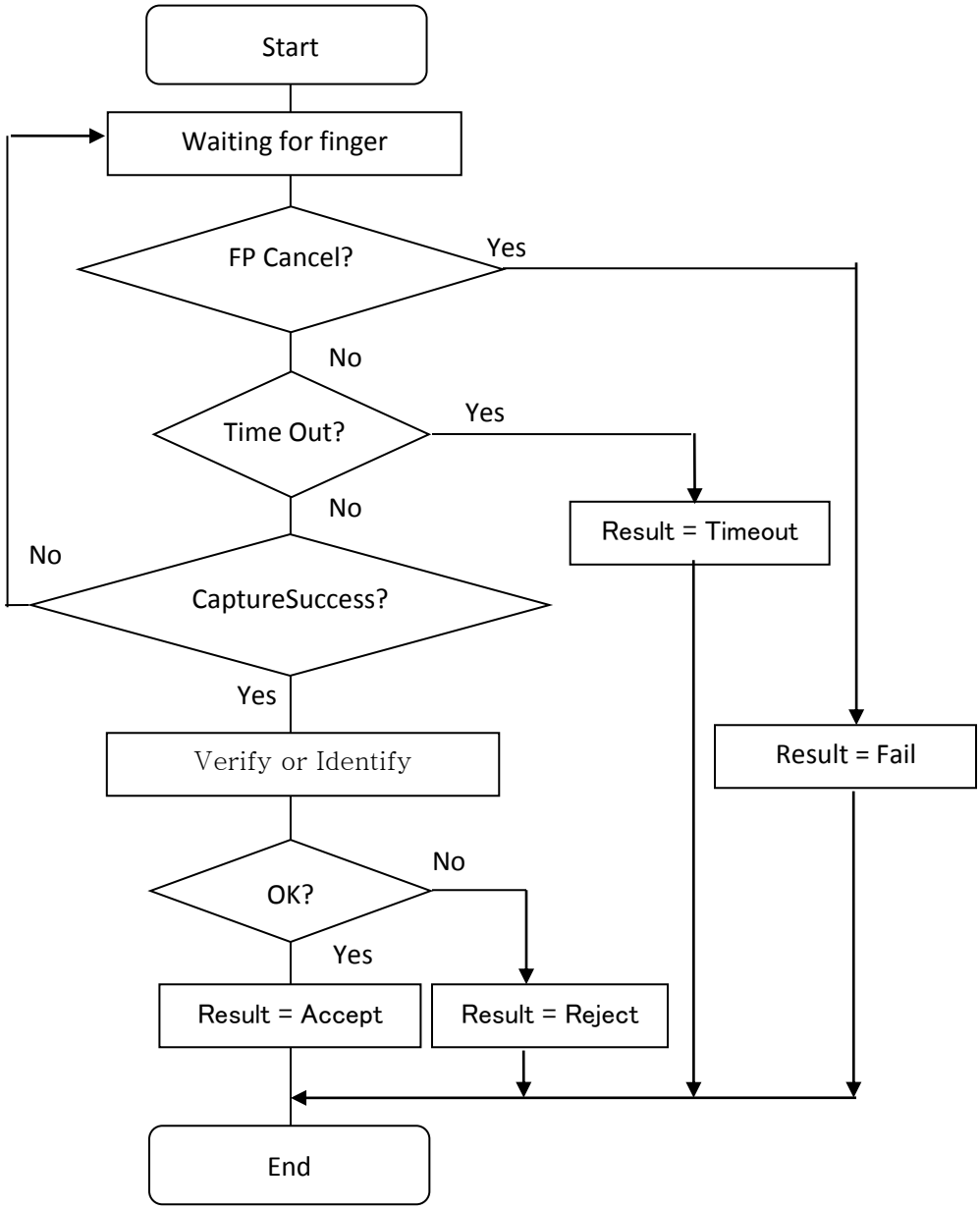


Fig 5-2 Flow of Verify and Identify

1N4001 - 1N400

1.0ARECTIFIER

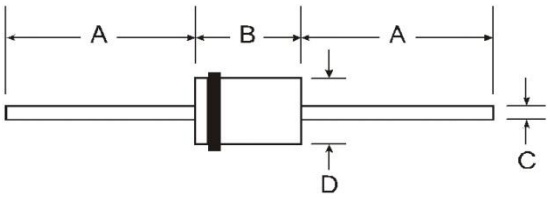
Please click here to visit our online spice models database.

Features

- Diffused Junction
- High Current Capability and Low Forward Voltage Drop
- Surge Overload Rating to 30A Peak
- Low Reverse Leakage Current
- **Lead Free Finish, RoHS Compliant (Note 3)**

Mechanical Data

- Case: DO-41
- Case Material: Molded Plastic. UL Flammability Classification Rating 94V-0
- Moisture Sensitivity: Level 1 per J-STD-020D
- Terminals: Finish - Bright Tin. Plated Leads Solderable per MIL-STD-202, Method 208
- Polarity: Cathode Band
- Mounting Position: Any
- Ordering Information: See Page 2
- Marking: Type Number
- Weight: 0.30 grams (approximate)



Dim	DO-41 Plastic	
	Min	Max
A	25.40	
B	4.06	5.21
C	0.71	0.864
D	2.00	2.72
All Dimensions in mm		

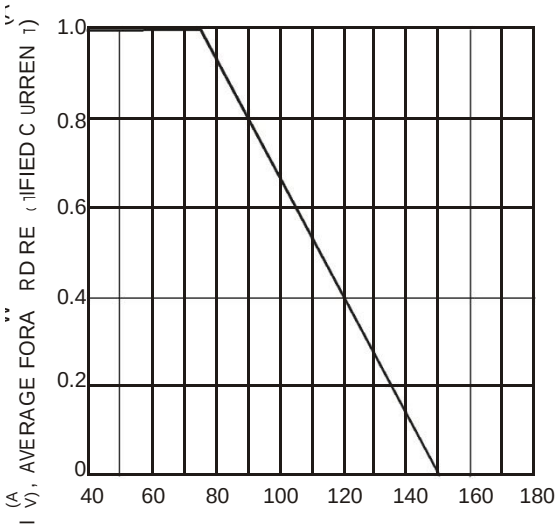
Maximum Ratings and Electrical Characteristics @T_A= 25°C unless otherwise specified

Single phase, half wave, 60Hz, resistive or inductive load.
For capacitive load, derate current by 20%.

Characteristic	Symbol	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	Unit
Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage	V_{RRM} V_{RWM} V_R	50	100	200	400	600	800	1000	V
RMS Reverse Voltage	$V_{R(RMS)}$	35	70	140	280	420	560	700	V
Average Rectified Output Current (Note 1) @ T_A = 75°C	I_O	1.0							A
Non-Repetitive Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load	I_{FSM}	30							A
Forward Voltage @ I_F = 1.0A	V_{FM}	1.0							V
Peak Reverse Current @T_A = 25°C at Rated DC Blocking Voltage @ T_A = 100°C	I_{RM}	5.0 50							μA
Typical Junction Capacitance (Note 2)	C_j	15				8			pF
Typical Thermal Resistance Junction to Ambient	$R_{\theta JA}$	100							K/W
Maximum DC Blocking Voltage Temperature	T_A	+150							°C
Operating and Storage Temperature Range	T_J, T_{STG}	-65 to +150							°C

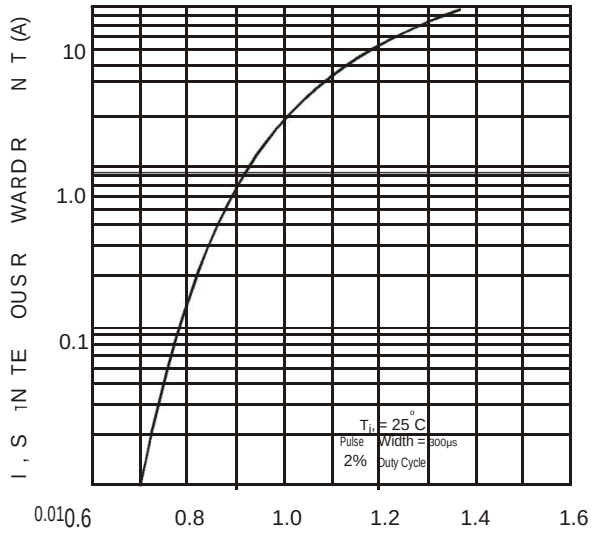
Notes:
1. Leads maintained at ambient temperature at a distance of 9.5mm from the case.
2. Measured at 1.0 MHz and applied reverse voltage of 4.0V DC.
3. EU Directive 2002/95/EC (RoHS). All applicable RoHS exemptions applied, see EU Directive 2002/95/EC Annex Notes.

© Diodes Incorporated



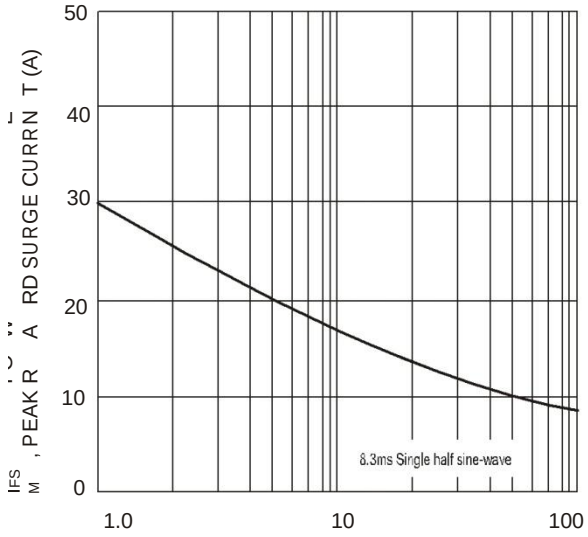
T_A, AMBIENT TEMPERATURE (ºC)

Fig. 1 Forward Current Derating Curve



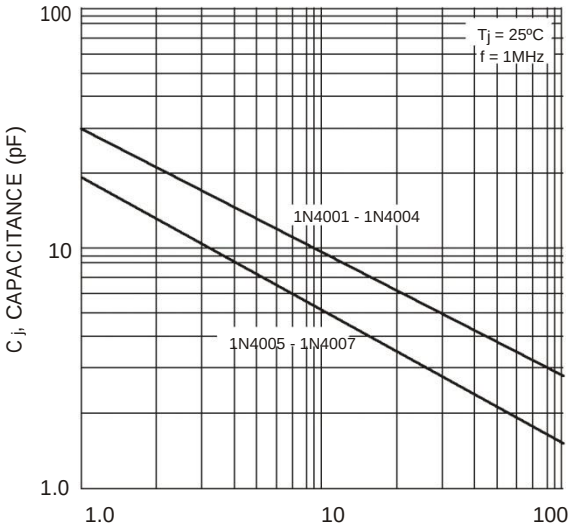
V_F, INSTANTANEOUS FORWARD VOLTAGE (V)

Fig. 2 Typical Forward Characteristics



NUMBER OF CYCLES AT 60 Hz

Fig. 3 Max Non-Repetitive Peak Fwd Surge Current



V_R, REVERSE VOLTAGE (V)

Fig. 4 Typical Junction Capacitance

© Diodes Incorporated

Ordering Information (Note 4)

Device	Packaging	Shipping
1N4001-B	DO-41 Plastic	1K/Bulk
1N4001-T	DO-41 Plastic	5K/Tape & Reel, 13-inch
1N4002-B	DO-41 Plastic	1K/Bulk
1N4002-T	DO-41 Plastic	5K/Tape & Reel, 13-inch
1N4003-B	DO-41 Plastic	1K/Bulk
1N4003-T	DO-41 Plastic	5K/Tape & Reel, 13-inch
1N4004-B	DO-41 Plastic	1K/Bulk
1N4004-T	DO-41 Plastic	5K/Tape & Reel, 13-inch
1N4005-B	DO-41 Plastic	1K/Bulk
1N4005-T	DO-41 Plastic	5K/Tape & Reel, 13-inch
1N4006-B	DO-41 Plastic	1K/Bulk
1N4006-T	DO-41 Plastic	5K/Tape & Reel, 13-inch
1N4007-B	DO-41 Plastic	1K/Bulk
1N4007-T	DO- 41 Plastic	5K/Tape & Reel, 13-inch

Notes: 4. For packaging details, visit our website at http://www.diodes.com/datasheets/ap02008.pdf.

IMPORTANT NOTICE

Diodes Incorporated and its subsidiaries reserve the right to make modifications, enhancements, improvements, corrections or other changes without further notice to any product herein. Diodes Incorporated does not assume any liability arising out of the application or use of any product described herein; neither does it convey any license under its patent rights, nor the rights of others. The user of products in such applications shall assume all risks of such use and will agree to hold Diodes Incorporated and all the companies whose products are represented on our website, harmless against all damages.

LIFE SUPPORT

Diodes Incorporated products are not authorized for use as critical components in life support devices or systems without the expressed written approval of the President of Diodes Incorporated.

© Diodes Incorporated

MICROCHIP

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
  DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

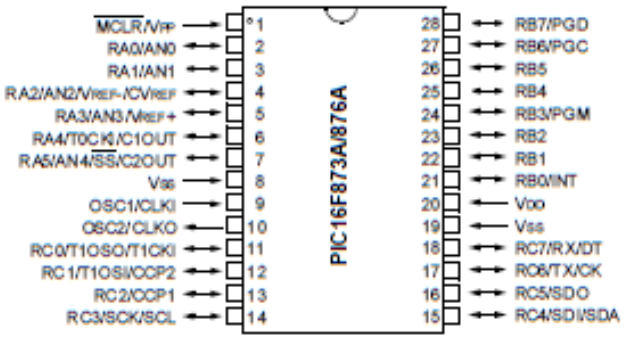
CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

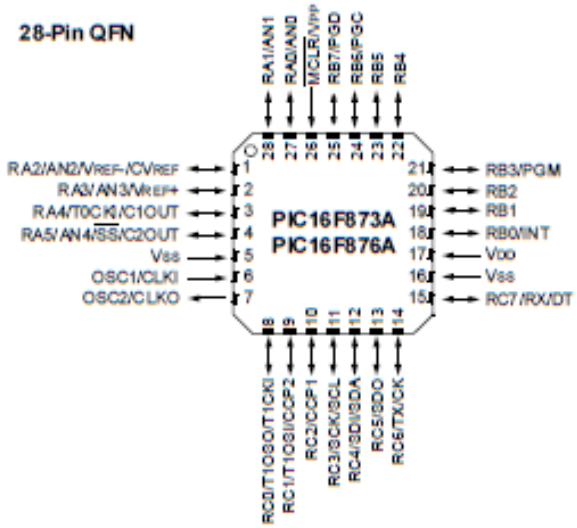
Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

Pin Diagrams

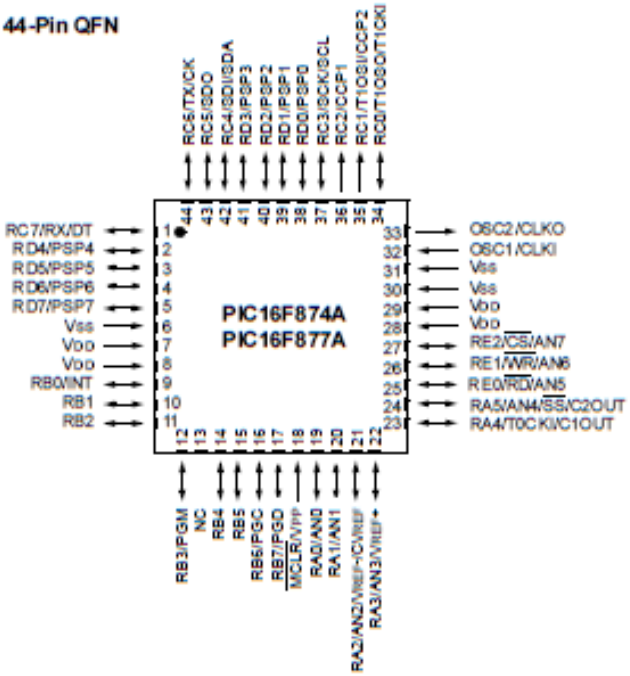
28-Pin PDIP, SOIC, SSOP



28-Pin QFN



44-Pin QFN



Pin Diagrams (Continued)